AD-A056 986 GEORGIA INST OF TECH ATLANTA F/G 17/2
DIGROUP DATA REDUCTION SYSTEM.(U)
JUN 78 J B O'NEAL F30602-75-C-0118
UNCLASSIFIED RADC-TR-78-139 NL

1 OF 3
AD
A056986

LEVEL II

(19) RADC-TR-78-139
Phase Report,
June 1978

(18) RADC

(12) 203p.

(6) DIGROUP DATA REDUCTION SYSTEM,

(10) J.B. O'Neal, Jr.

North Carolina State University

(15) F30602-75-C-0118

Approved for public release; distribution unlimited

(16) 9567 (17) 00

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

153 800

This report contains a large percentage of copy which is not of the highest printing quality but because of economical consideration, it was determined in the best interest of the government that they be used in this publication.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-139 has been reviewed and is approved for publication.

APPROVED: Jacob Scherer
JACOB SCHERER
Project Engineer

APPROVED: Joseph J. Naresky
JOSEPH J. NARESKY
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBC) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-78-139	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DIGROUP DATA REDUCTION SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Phase Report
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) J.B. O'Neal, Jr.		8. CONTRACT OR GRANT NUMBER(s) F30602-75-C-0118
9. PERFORMING ORGANIZATION NAME AND ADDRESS North Carolina State University Raleigh NC 27650		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ~~920W~~ 95670018
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBC) Griffiss AFB NY 13441		12. REPORT DATE June 1978
		13. NUMBER OF PAGES 206
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Jacob Scherer (RBC)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

speech transmission
digital encoding
adaptive differential PCM
speech compression

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the design and construction of a Digroup Data Reduction (DDR) system. The DDR system compresses the bit rate of two TI PCM telephone speech carrier systems down into a 1.544 mb/sec bit stream normally occupied by one Tl carrier system. The DDR system uses a combination of Differential PCM (DPCM) and Time Assignment Speech Interpolation (TASI) to accomplish this 2:1 compression task. The report contains design details of the prototype model. The prototype was constructed using standard TTL logic.

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This effort was conducted by North Carolina State University under the sponsorship of the Rome Air Development Center Post-Doctoral Program. Mr. John Stacy of RADC was the task project engineer and provided overall technical direction and guidance. J.B. O'Neal, Jr. directed this research and the preparation of this report at NC State University. The authors of the report are J.B. O'Neal, Jr., G.A. Williamson, T.R. McPherson, R.P. Gooch, R.B. Martin and M.W. Howard.

The RADC Post-Doctoral Program is a cooperative venture between RADC and some sixty-five universities eligible to participate in the program. Syracuse University (Dept. of Electrical Engineering), Purdue University (School of Electrical Engineering), Georgia Institute of Technology (School of Electrical Engineering), and State University of New York at Buffalo (Dept. of Electrical Engineering) act as prime contractor schools with other schools participating via sub-contracts with the prime schools. The U.S. Air Force Academy (Dept. of Electrical Engineering), Air Force Institute of Technology (Dept. of Electrical Engineering), and the Naval Post Graduate School (Dept. of Electrical Engineering) also participate in the program.

The Post-Doctoral Program provides an opportunity for faculty at participating universities to spend up to one year full time on exploratory development and problem-solving efforts with the post-doctorals splitting their time between the customer location and their educational institutions. The program is

totally customer-funded with current projects being undertaken for Rome Air Development Center (RADC), Space and Missile Systems Organization (SAMSO), Aeronautical Systems Division (ASD), Electronic Systems Division (ESD), Air Force Avionics Laboratory (AFAL), Foreign Technology Division (FTD), Air Force Weapons Laboratory (AFWL), Armament Development and Test Center (ADTC), Air Force Communications Service (AFCS), Aerospace Defense Command (ADC), HQ USAF, Defense Communications Agency (DCA), Navy, Army, Aerospace Medical Division (AMD), and Federal Aviation Administration (FAA).

Further information about the RADC Post-Doctoral Program can be obtained from Jacob Scherer, RADC/RBC, Griffiss AFB, NY, 13441, telephone AV 587-2543, COMM (315) 330-2543.

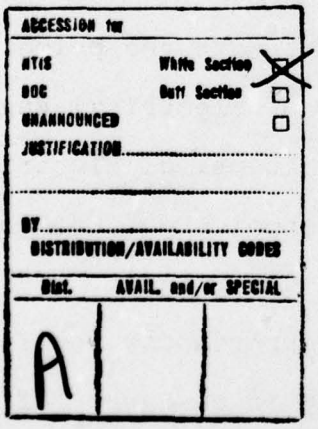


TABLE OF CONTENTS

LIST OF FIGURES

* The original drawings for these figures have been reduced for publication in this report. These figures may therefore be difficult to use. Full-scale drawings of these figures are available from J.B. O'Neal, Jr., NC State University, P.O. Box 5275, Raleigh, NC, 27650.

1. INTRODUCTION

The purpose of this report is to document the design and construction of a prototype model of a Digroup Data Reduction (DDR) system. This system compresses the bit rate of two T1 PCM carrier terminals down into the bit rate normally required by one T1 carrier terminal. Each T1 carrier terminal encodes 24 speech channels into a 1.544 Mb/s digital signal which is transmitted over a T1 repeatered line. The DDR system allows two T1 systems (48 speech channels) to operate over one T1 repeatered line. Fig 1.1 shows the operation of T1 carrier systems with and without compression by the DDR.

The DDR system described here uses a combination of Adaptive Differential Pulse Code Modulation (ADPCM) and Time Assignment Speech Interpolation (TASI). Compression systems of this type are called ADPCM/TASI. A previous report [1] describes a study done to determine which compression techniques seemed best suited to the DDR application. That study describes the theoretical work and computer simulation upon which the decision was made to use ADPCM/TASI for the DDR application. The current report describes the design and construction of a prototype model of this ADPCM/TASI system.

The concept of ADPCM is well known but its use with TASI has not been previously explored in any detail [1]. The TASI system itself has long been used on undersea analog cable transmission systems [2]. Two characteristics of speech enable the DDR system to perform the compression: (1) the user of a two-

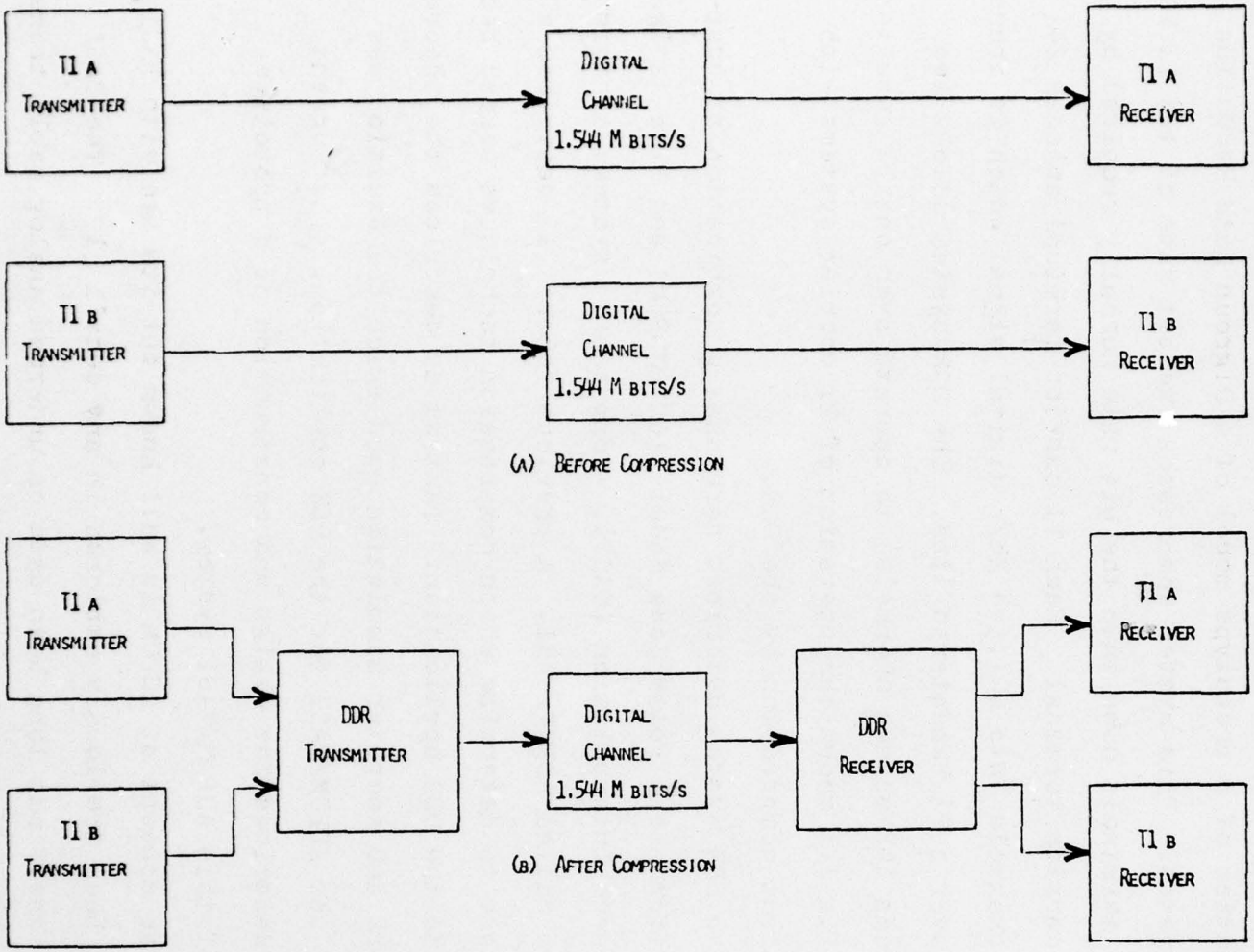


Figure 1.1 Speech Compression in a T1 Environment
for One Direction of Transmission

way communication system talks only about 38% to 45% of the time the system is off-hook--the duty factor is .38 to .45--and (2) most sequences of speech samples are slowly varying compared to the PCM sampling rate and are therefore somewhat predictable. There are times when the system is heavily loaded and more than 38% to 45% of the users in one direction are talking. ADPCM/TASI adapts to high activity by reducing the number of bits used to encode each sample value. By reducing the number of bits during periods of heavy loading, the DDR system degrades gracefully--all active channels get a time slot. This is significantly different from systems such as TASI or Speech Predictive Encoding Communications (SPEC), where high activity results in certain channels being frozen out or updated infrequently.

Since the DDR system is tailored to the statistics of speech, it may not operate well with non-speech signals such as voiceband data. The DDR system considered was arranged so that up to 6 of the 48 time-division-multiplexed T1 channels could pass through without modification. The time slots assigned to these channels would bypass the ADPCM/TASI operations and could therefore be assigned to data. This way of handling the data limits the utility of this system in networks where voiceband data and speech coexist on voicegrade channels.

Hardware implementation of the DDR system consisted of breaking the tasks into natural functional divisions, interfacing the functional blocks, construction, trouble-shooting, and documentation. The heart of the DDR hardware system consists of the

voice switch and ADPCM coder. The voice switch detects the presence of speech in order to take advantage of the low duty factor of speech by using the TASI concept. In turn, the ADPCM coder takes advantage of the predictability of speech in order to reduce the number of required bits.

The performance of the hardware DDR system--signal-to-noise ratios under various dynamic loading configurations and formal subjective tests--has not yet been evaluated; however, computer simulations of the concept showed no noticeable degradation.

2. THEORETICAL ASPECTS

2.1 Pulse Code Modulation (PCM)

The advantages of PCM are well known. Briefly, they are: (1) PCM signals may easily be reshaped or regenerated in a noisy environment-the effect of noise in the transmission medium can be almost eliminated, (2) economical digital circuitry may be used, (3) easy encryption, (4) ease of multiplexing via time-division-multiplexing, and (5) noise may be minimized by appropriate coding of the signal. The crux of the matter is that high signal-to-noise ratios can be maintained on communication channels with low signal-to-noise ratios. These advantages are obtained at the cost of increased bandwidth requirements for the transmission channel.

In the T1 environment, speech or voiceband data signals that are bandlimited to approximately 3.2 KHz are sampled at slightly over the Nyquist rate--8000 samples/s. Then the samples are logarithmically quantized at 8 bits/sample, using a μ255 quantizer. The purpose of the logarithmic quantizer is to give the PCM coder a constant signal-to-quantizing noise ratio (S/N_q) over a wide range of signal levels. For a further discussion of logarithmic quantizers, see [5,6]. In addition to performing the PCM coding, the T1 system time-division multiplexes 24 speech or voiceband data channels into a single communications channel, resulting in a bit rate of 1.544 M bits/s. A typical T1 frame is shown in Figure 2.1.

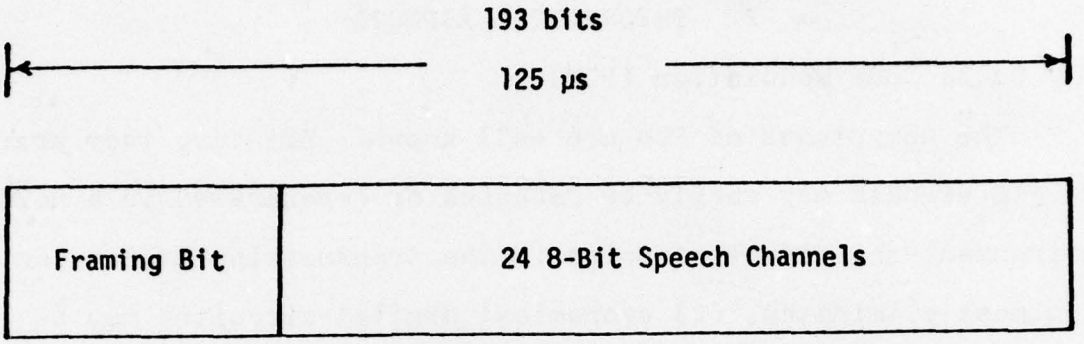


Figure 2.1 The T1 Frame.

The sampling theorem indicates that a bandlimited signal can be reproduced perfectly at the receiver if sampled at a rate of twice the signal bandwidth. In any practical situation, the signal can not be perfectly bandlimited and the samples can not be represented with perfect accuracy. The latter problem usually dominates; hence, it is useful to consider the signal-to-quantizing noise ratio (S/N_q) as a performance criterion. For an n-bit $\mu 255$ quantizer, the signal-to-quantizing noise ratio is simply [6]:

$$S/N_q = 6n - 10.1 \quad \text{dB}.$$

The constant, 10.1 in this equation depends on the companding rule. For the 8-bit T1 quantizer,

$$S/N_q = 6(8) - 10.1 = 37.9 \quad \text{dB}.$$

2.2 Adaptive Differential Pulse Code Modulation (ADPCM)

It is well known that sample-to-sample values of PCM encoded speech are correlated. This property allows speech samples to be predicted. Unfortunately, the autocorrelation function of

speech waveforms is non-stationary; however, the time variation is not large and the assumption of stationarity leads to the design of useful systems. A typical sampled autocorrelation function for speech at sampling intervals of 125 μsec. is given in Table 2.1 [7].

Table 2.1 Typical Autocorrelation Function for Speech

i	R(i)
0	1
1	.866
2	.554
3	.225

The motivation for using DPCM can easily be seen from the following calculations. Consider the DPCM encoder shown in Figure 2.2. Suppose the s_i are input speech samples with the autocorrelation function of Table 2.1. The linear predictor is of the form

$$s_i = \sum_{j=1}^{m} \alpha_j \, s_{i-j}$$

neglecting quantizing noise, q_i, in the feedback loop. For telephone quality signal-to-noise ratios, $s_i >> q_i$. The purpose of the prediction is to make the variance of $(s_i - \hat{s}_i)$ smaller than that of s_i so that a lesser number of bits will be required for quantization, i.e., minimize

$$\sigma_{d_i}^2 = E\{(s_i - \hat{s}_i)^2\}$$
$$= E\{(s_i - \sum_{j=1}^{m} \alpha_j \, s_{i-j})^2\}$$

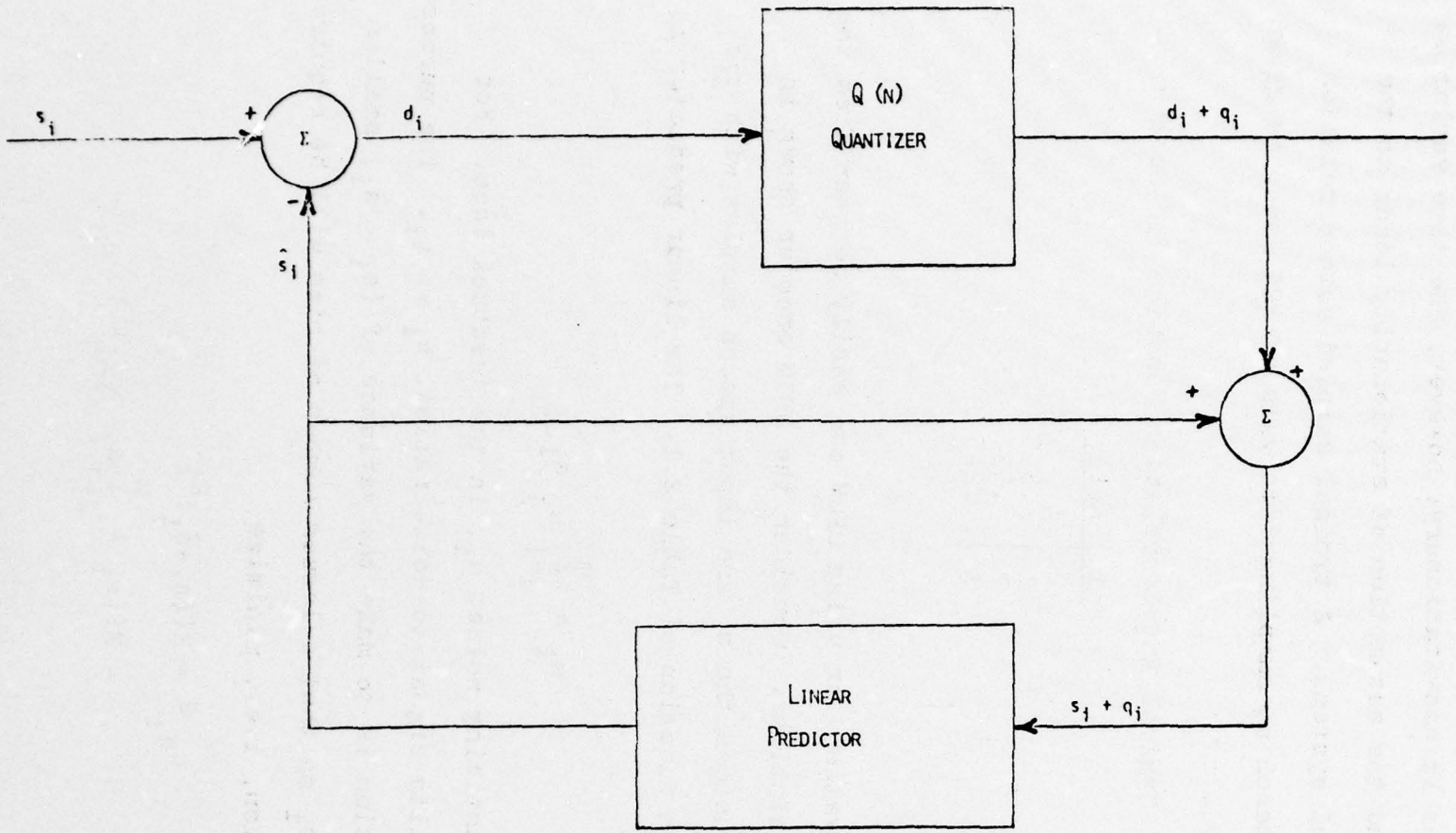


Figure 2.2 General DPCM Encoder

by choosing the appropriate α_j. This minimization can easily be done by taking the partial derivative of $\sigma_{d_i}^2$ with respect to α_j for j = 1 to m, setting the resulting equations equal to zero, and then solving the m equations for m unknowns. Carrying out this procedure for m = 1,2,3 gives the α_j of Table 2.2.

Table 2.2 Predictor Coefficients for Speech

Predictor	α_1	α_2	α_3
one-tap	.866		
two-tap	1.545	-.7837	
three-tap	1.936	-1.553	.4972

The signal-to-quantizing noise ratio can be expressed as:

$$S/N_q = E\{s_i^2\}/E\{q_i^2\} = 1/\sigma_{d_i}^2 Q(n)$$

where $Q(n) = \sigma_{q_i}^2/\sigma_{d_i}^2$ is a characteristic of the quantizer. If the predictor were not present, the sequence s_i would be quantized since $s_i = d_i$ in this case and the signal-to-quantizing noise ratio would be

$$S/N_q = 1/Q(n).$$

Hence the signal-to-noise ratio improvement, SNI, from prediction is

$$SNI = -10 \log \sigma_{d_i}^2 \quad dB.$$

Table 2.3 gives the SNI for the three predictors described above.

Table 2.3 SNI for Various DPCM Predictors with Speech Input

Predictor	SNI (dB)
one-tap	6.02
two-tap	10.2
three-tap	11.4

The improvement from one-tap to two-tap is not large and, as a practical matter, is further reduced by the nonstationarity of the speech signal. Also the three-tap predictor is very sensitive to variations in R(i) and would require much more hardware than the one-tap predictor. Hence, the one-tap predictor was picked for the DDR system. Furthermore, an $\alpha_1 = 0.875$ is used for even more simplicity. Since $1 - (1/8) = 0.875$, the multiplication only requires a shift right and subtract. Figure 2.3 shows the resulting ADPCM encoder and decoder.

The SNI for this coder can be calculated using Table 2.1:

$$\sigma_{d_i}^2 = E\{(s_i - \alpha_1 s_{i-1})^2\}$$

$$= (1 + \alpha_1^2)\, E\{s_i^2\} - 2\alpha_1 E\{s_i s_{i-1}\}$$

$$= (1 + \alpha_1^2)\, R(0) - 2\alpha_1 R(1)$$

$$= 0.250$$

therefore,

$$SNI = -10 \log (\sigma_{d_i}^2)$$

$$= 6.02 \quad dB .$$

The quantizer used in the ADPCM encoder is fashioned after

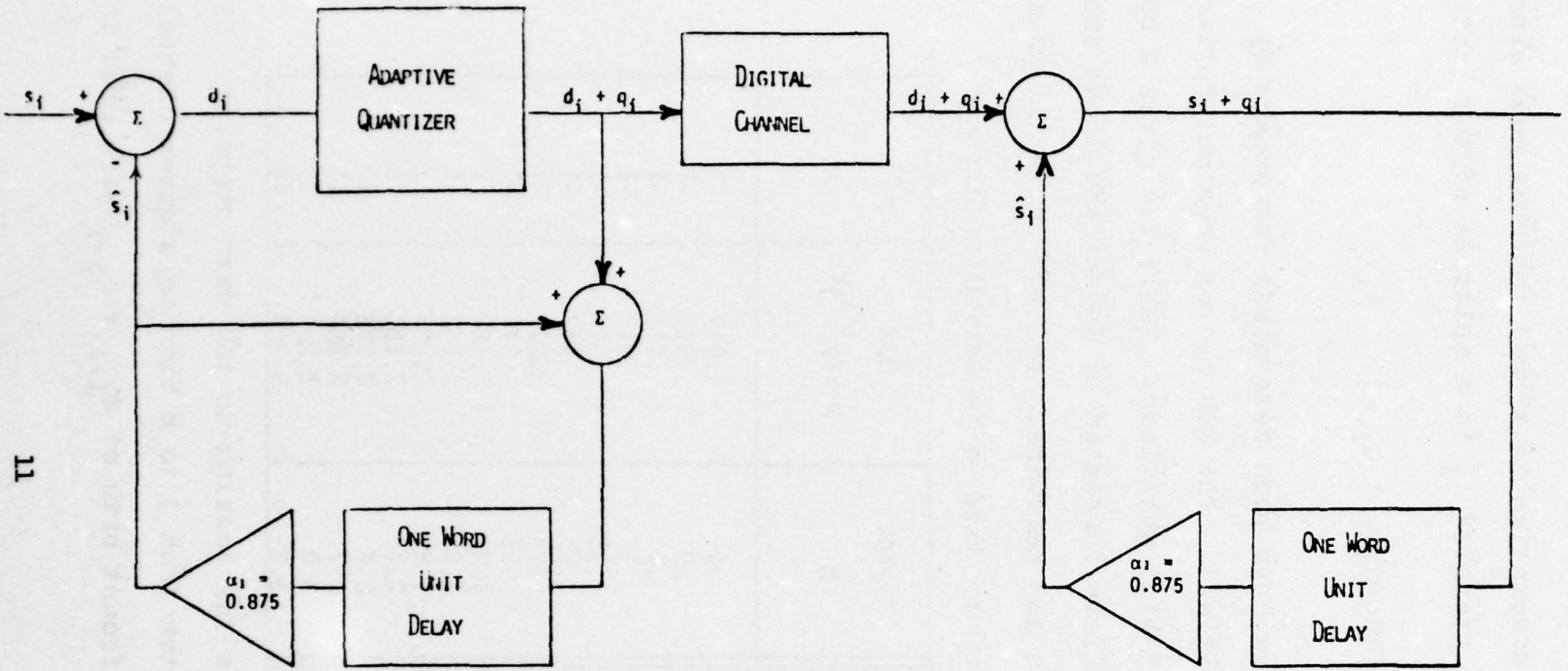


Figure 2.3 ADPCM Encoder and Decoder for a Speech Channel

that of Jayant [8]. Adaptation is termed instantaneous since the current quantizer stepsize Δ_i is a multiple of the last; i.e.,

$$\Delta_i = M(\text{abs}(d_{i-1} + q_{i-1})) \cdot \Delta_{i-1}.$$

The optimum multipliers have been determined for speech by Jayant and are not critical. The DDR system uses multipliers based on a 32 level (5-bit) quantizer. Table 2.4 gives a comparison of the multipliers suggested by Jayant and those used in the DDR simulations and hardware. The $2^{1/8}$ rule allows a

Table 2.4 Multipliers Used in the Adaptation Strategy

$\lvert d_{i-1} + q_{i-1} \rvert$	Jayant M	DDR $M = (2^{1/8})^k$	k
0000	.9	.917	-1
0001	.9	.917	-1
0010	.9	.917	-1
0011	.9	.917	-1
0100	.95	.917	-1
0101	.95	.917	-1
0110	.95	.917	-1
0111	.95	.917	-1
1000	1.2	1.189	2
1001	1.5	1.542	5
1010	1.8	1.834	7
1011	2.1	2.181	9
1100	2.4	2.378	10
1101	2.7	2.594	11
1110	3.0	3.084	13
1111	3.3	3.364	14

simple indexing scheme for looking-up the next stepsize. Since the DDR system allocates from 3 to 8 bits to a given channel, only the 5 most significant bits of $d_{i-1} + q_{i-1}$ were used in the

simulations. The hardware system used a variable number of bits to look-up multipliers that are an interpolated version of Table 2.4. For further details of the hardware quantizer, see chapter 4.

Performance of the ADPCM coder was evaluated by simulation and is in [1]. At this point, it is worth noting that Jayant's results suggest an S/N_q of 27 dB for ADPCM with a 5-bit quantizer (Ref. 8). This is slightly below telephone quality; however, the DDR system should usually allocate more than 5 bits, as seen in Section 2.5, resulting in an S/N_q in the 30 dB plus range. Subjective tests indicate even better results as discussed in [1].

2.3 Code Conversion

The ADPCM coder considered above operates on a uniformly quantized input signal; however, the T1 system outputs logarithmically quantized sample values. An obvious solution to the problem is some sort of code conversion. The μ255 quantizer uses a 15 segment linear approximation to the following companding function (Ref. 5):

$$F(x) = \text{sign}(x)\,\frac{\ln(1 + \mu\,\text{abs}(x))}{\ln(1 + \mu)} \qquad -1 \le x \le 1,\ \mu = 255.$$

The DDR code conversion device must perform this function and its inverse on a finite set of digital input values. Table 2.5 gives the conversion rule where n is the 8-bit T1 output, $\hat{y}$ is the linearly quantized speech sample at the encoder, y is the reconstructed linearly quantized speech sample at the decoder, and

Table 2.5 Code Conversion Table (Ref. 6)

8 bits μ=255

y + 33	n	$\hat{y}$ + 33
1wxyzabcdefgh	111wxyz	1wxyz10000000
01wxyzabcdefg	110wxyz	01wxyz1000000
001wxyzabcdef	101wxyz	001wxyz100000
0001wxyzabcde	100wxyz	0001wxyz10000
00001wxyzabcd	011wxyz	00001wxyz1000
000001wxyzabc	010wxyz	000001wxyz100
0000001wxyzab	001wxyz	0000001wxyz10
00000001wxyza	000wxyz	00000001wxyz1

e is the error incurred in the transmission process, if any. The 33 is added to the y and $\hat{y}$ entries in order to simplify the table. Figure 2.4 shows the place of code conversion in the DDR system. The 8-bit μ255 to 14-bit uniform conversion can easily be implemented using a PROM; whereas, the reverse process requires special purpose digital circuitry. More details on code conversion hardware are given in Chapter 4.

2.4 Time Assignment Speech Interpolation (TASI)

The basic idea of Time Assignment Speech Interpolation (TASI), sharing of c transmission channels among n off-hook users, is not new (Ref. 2 and Ref. 9). It has been used for many years on undersea cable. TASI takes advantage of the fact that a user of a communications channel talks, on the average, only about 38% to 45% of the time. For the remainder of the time, the user

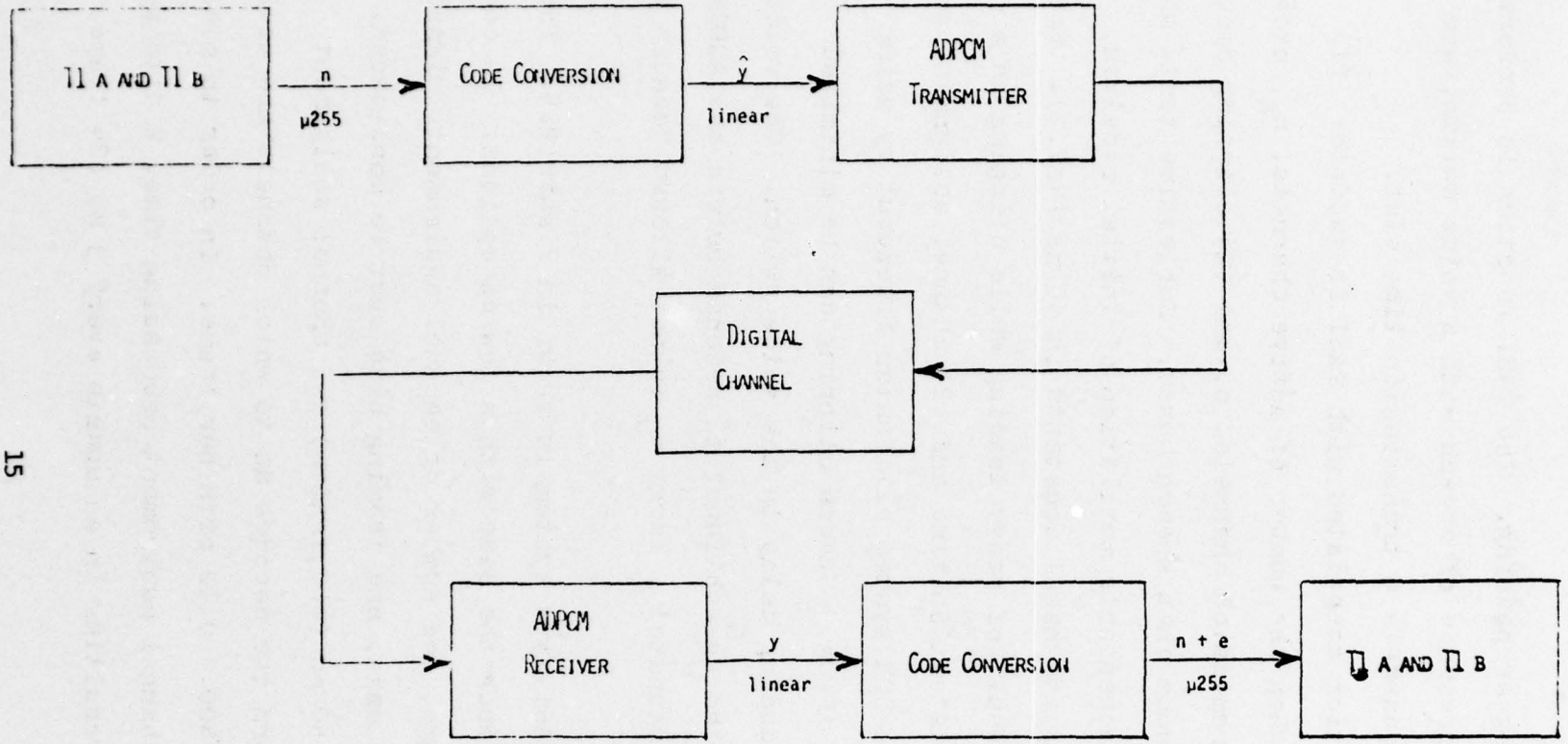


Figure 2.4 Code Conversion
in the DDR System

is either listening or pausing. The TASI function is performed by detecting the presence of speech with a voice switch, and assigning active channels a transmission time slot.

The degradation associated with TASI is twofold: (1) freezeout occurs when the number of active channels, n_A, exceeds the number of transmission channels, c, and (2) clipping occurs on the initial sounds in a speech burst. Subjective tests show that the maximum acceptable mutilation of initial plosives, stops, fricatives, and nasal constants is 50 ms (Ref.3). Freeze-out occurs as a result of heavy loading while clipping is a result of : (1) detection time and (2) channel assignment processing time. The DDR system eliminates freezeout by using variable wordsize ADPCM. Speech clipping can be minimized somewhat by introducing delay in the voice switch. The voice switch processes the speech input t_D seconds before the ADPCM/TASI coder. Unfortunately, large t_D is not allowed because of echo problems.

A general TASI-type system is shown in Figure 2.5. The DDR system looks much the same with a few exceptions. In constructing the frame, the number of channel assignment, signaling, speech, voiceband data, and framing bits must be considered. At least once every 50 ms (400 frames) the channel assignment message must inform the decoder as to which channels are active. This requires 48/400 = 0.12 bits per frame. In order to minimize clipping due to channel assignment processing time, 2 bits per frame are sent, resulting in an update every 3 ms (24 frames).

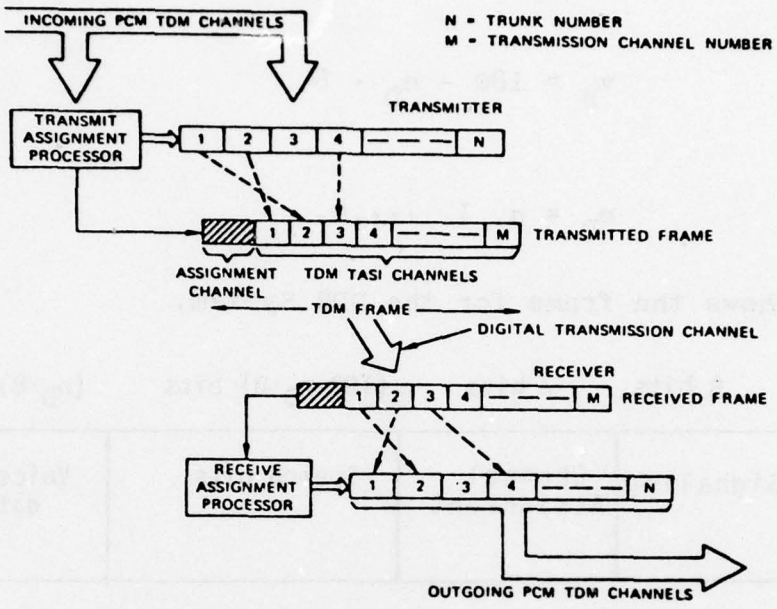


Figure 2.5 A General TASI System (Ref. 3)

Since it is imperative that the DDR transmitter and receiver have the same channel assignment information, an error correction code of rate $\frac{1}{2}$ is used to protect the channel assignment message. Four information bits are sent, followed by four parity bits at the rate of four bits per frame. See Chapter 4 for further details of the error correcting code. In a T1 system, signaling is sent in the least significant bit of the 8-bit μ255 word every sixth frame. Hence, 48/6 = 8 bits per frame must be allocated to signaling. One bit per frame is required for framing as in the T1 system and the remaining bits are shared between the n_D voiceband data and n_A active voice channels. Since the standard T1 frame has 193 bits, the bits available for

voice is

$$v_B = 180 - n_D \cdot 8$$

where

$$n_D = 0, 1, \cdots, 6.$$

Figure 2.6 shows the frame for the DDR System.

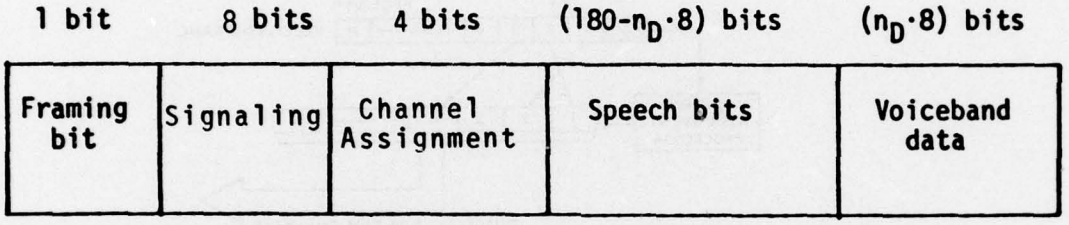


Figure 2.6 The DDR Frame.

A critical part of any TASI system is a voice switch that can detect the presence of speech in a noisy environment. Many algorithms have been devised for detecting speech; one in particular is well suited to the DDR application (Ref. 10). It uses level detection and adaptive thresholds operating on μ255 companded PCM. This switch shows no noticeable speech clipping, high immunity to false triggering, and fast adaptation to changes in noise and speech levels. For further details of the algorithms used and hardware switch, see Chapter 4.

2.5 Talker Statistics and Bit Allocation

Past studies have shown that the average speech activity factor, α, is in the range 0.38 to 0.45 and that the mean talkspurt length--the time the voice switch is on--is exponentially

distributed with a mean, S, of approximately 1.5 s (Ref. 2, 3, 9, and 11). The traffic in the DDR simulations was generated using these properties. Stress on the DDR system for various loading conditions is difficult to estimate quantitatively; however, some indication of the loading can be obtained if the percent of time B bits is available is known. This can be calculated given the activity factor and loading configuration.

At any particular instant, the probability that the number of simultaneous talkers will equal or exceed c (where c is less than n) is given by the cumulative binomial expansion:

$$B(c, n; \alpha) = \sum_{x=c}^{n} \frac{n!}{x!(n-x)!} \alpha^x (1-\alpha)^{n-x}$$

where n is the total number of off-hook talkers and α is the speech activity factor (Ref. 9). Before using this to calculate the fraction of time B bits are available, Φ_B, bit allocation must be discussed. The maximum number of channels for which B bits will be available is given by

$$c_B = \langle v_B/B \rangle$$

where

$$v_B = 180 - n_D \cdot 8$$

as discussed in Section 2.4. $\langle \cdot \rangle$ means integer part of. For a given number of active channels n_A and available voice bits v_B, two positive integers, k_1 and k_2, must be found such that

$$k_1 \cdot L + k_2 (L+1) = v_B$$

in order to use all available bits. L is given by

$$L = \langle v_B/n_A \rangle .$$

Once this computation is done, bit allocation consists of giving the first k_1 active channels in a frame L bits and the last k_2 active channels L + 1 bits. On the average, channels toward the end of the frame will receive more bits than those at the beginning. Given a channel counter, k, that counts the active channels in a given frame period, the number of bits available, B, can be written symbolically as

$$B = \begin{cases} L & k \leq k_1 \\ L+1 & k > k_1 \\ 8 & \text{Selected data channels.} \end{cases}$$

In the context above, channel 1 would receive L bits whenever active. Now Φ_B for channel 1 can easily be calculated for various configurations. Two cases must be considered: (1) 8 bits are available as long as c_8 or less channels are active, implying $\Phi_8 = 1 - B\ (c_8 + 1,\ 48 - n_D;\ \alpha)$ and (2) B bits are available, where $3 \leq B \leq 7$, as long as $c_{B+1} < n_A \leq c_B$ implying $\Phi_B = B(c_{B+1} + 1,\ 48 - n_D;\ \alpha) - B\ (c_B + 1,\ 48 - n_D;\ \alpha)$.

To show the effects of various loading configurations in the DDR system, three cases have been computed: (1) full loading with no data channels and a low activity factor; $n = 48$, $n_D = 0$, $\alpha = .38$, (2) full loading with some data channels and a moderate activity factor; $n = 48$, $n_D = 4$, $\alpha = .4$, and (3) full loading with all data channels and a high activity factor; $n = 48$,

$n_D = 6$, $\alpha = .45$. Table 2.6 gives the resulting ϕ_B. The large variation for various loading configurations points to the fact that an effective bit dropping scheme like variable word length ADPCM is needed. Jayant indicates that his 5-bit ADPCM system is subjectively equivalent to 7-bit PCM (Ref. 12). Table 2.6 indicates that even under worst case conditions, 5 or more bits will be available 99.08% of the time. This means that the DDR system will perform as well as 7-bit PCM even under worst case conditions. Informal listening tests of speech from computer simulations of the DDR system have verified this. These computer simulations are covered in (Ref. 1).

Table 2.6 Fraction of Time B bits are Allocated for Various Loading Configurations

B	n=48 n_D=0 α=.38	n=48 n_D=4 α=.40	n=48 n_D=6 α=.45
8	0.8964	0.6126	0.2293
7	0.0870	0.2718	0.2235
6	0.0164	0.0979	0.4150
5	0.0002	0.0176	0.1230
4	0.	0.0001	0.0092
3	0.	0.	0.

3. DDR SYSTEM CONSIDERATIONS

3.1 Design

The DDR system is designed to accept the outputs of and provide the inputs to two T1 carrier systems, as shown in Figure 1.1. A T1 carrier system is a 24 channel PCM telephone system widely used in long-distance commercial and military communications. Since the operation of the T1 system necessarily influences the DDR design, a description of its operation is required.

The T1 system accepts 24 telephone channels for transmission. Each 4 kHz line is sampled 8000 times a second. Each sample is quantized and encoded into an 8-bit, companded, signed magnitude word. These 24 8-bit words plus a framing bit for synchronization are combined to form a 193-bit frame as shown in Figure 3.1. Eight thousand 193-bit frames are transmitted each second yielding a bit rate of 1.544 M bit/sec. Every sixth frame the least significant bit of each channel is used for signaling (dialing, on/off hook, etc.). This T1 frame format is both the input to the DDR encode and the output of the DDR decode.

Several other properties of the T1 output must be considered. The repeater clocks are driven by the "ones" on the transmission line. A string of more than 14 zeros will allow the clock to drift excessively. The T1 solves this problem by preventing eight "ones" in any channel and

inverting the output. This allows a maximum of 14 zeros on the line (eq., ...1000 0000 0000 0001...).

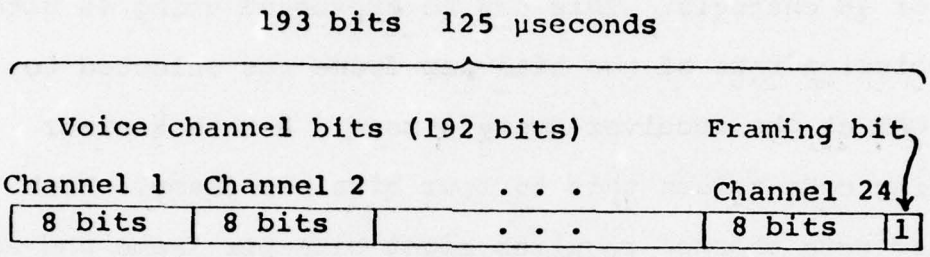


Figure 3.1 The T1 carrier frame

The input to the T1 system is normally 24 channels of speech. However, it is possible that digital data will occur on the T1 channels — this data could be PCM encoded modem signals or it could originate as strictly digital data. Any signal of this type will not accept compression. The DDR system will require that this type of input appear only on no more than six designated channels. These channels when so designated will be handled as digital data and will bypass the ADPCM/TASI algorithm.

Finally, there is no requirement that two independent T1 systems be bit synchronous. However, to allow asynchronous inputs to the DDR would require extensive buffering. To eliminate this problem, the two T1 systems are required to be bit synchronous. This basically defines the DDR inputs and requirements.

The DDR must transmit signaling, channel assignment, voice data, dedicated data, and a framing bit. Fourty-eight

bits of signaling are received every sixth frame. The DDR must send eight bits per frame to send all the bits. Channel Assignment Information (CAI) consists of the on/off status of 48 channels. This can be expressed using 48 bits. A transmission rate of two bits per frame was selected to update CAI at the receiver every 3 msec. A rate ½ error correcting code raises this to four bits per frame. Each dedicated data channel requires eight bits per frame and each frame requires one framing bit. The rest is used for voice data. This DDR frame is shown in Figure 3.2.

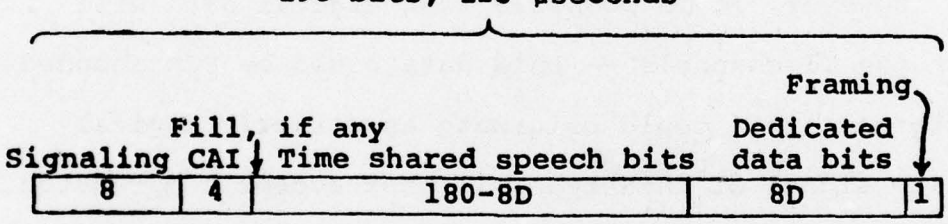


Figure 3.2 The DDR frame

Each 193-bit frame has 180 bits available for data. If D is the number of dedicated data channels, 8D bits are used for dedicated data and 180-8D bits are used for voice data. The DDR shares the 180-8D bits among the up to 48-D active channels. For instance, with three dedicated data channels and 20 active voice channels, the 156 bits are assigned seven each for the first four channels and eight each for the remaining 16. When there are more than eight bits available per active channel, fill bits are inserted between the CAI bits and the voice data bits. The DDR frames are

transmitted 8000 times a second to yield a bit rate of 1.544 bits/sec, the same as the T1's.

A block diagram of one direction of the DDR system is shown in Figure 3.3. Each block is described in some detail below.

The inputs to the DDR encoder are the bit streams of two T1 carrier terminals labeled T1A and T1B. This system requires the two transmitters to be bit synchronous, a minor modification of the standard T1 system. This requirement may also impose certain physical limitations on the allowable separation of the T1 terminals; i.e., the physical length of a sync wire between them.

The output of T1A enters a modified T1 receive card labeled Receive A. This card establishes frame and decodes the serial bit stream into 8-bit parallel words. It also provides the input to the Sync and Timing block which extracts the required synchronization and timing for the entire DDR encode section.

The output of T1B enters another modified receive card labeled Receive B. Although this input is bit synchronous with T1A, it is probably not frame synchronous with T1A. The De-Skew Buffer is added to align the frames of the two inputs. This block detects any misalignment and delays the B signal until the frames are aligned. The Combiner receives 8-bit parallel words from both A and B Receive and interlaces them into a 48-channel frame. Each 8-bit

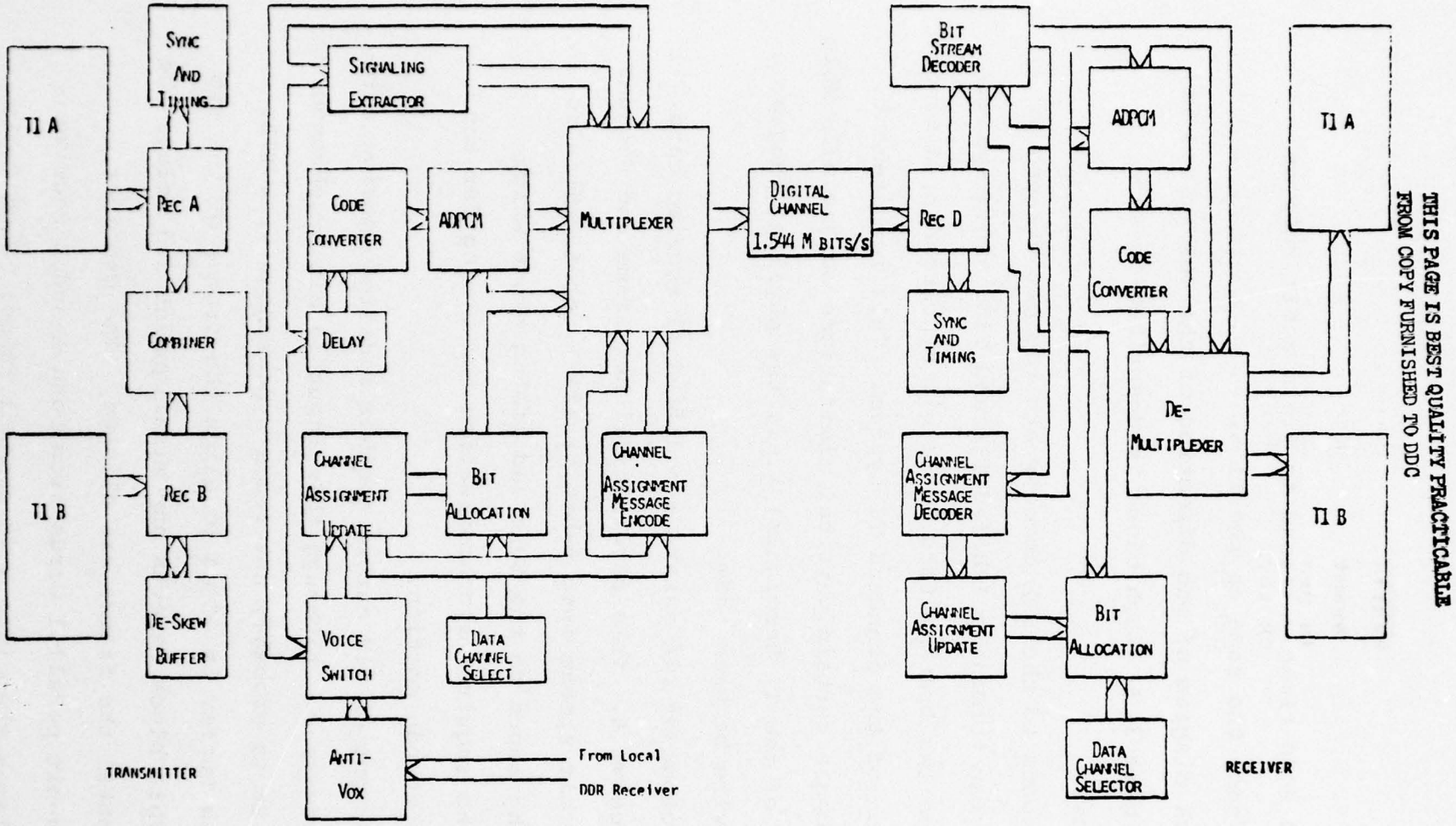


Figure 3.3 Hardware Oriented Block Diagram of the DDR System

parallel word is now available for 2.6 μsec to the rest of the encoder.

The voice switch examines each 8-bit word to determine whether or not voice is present on each of the 48 channels. The Echo Suppress block receives its input from the local DDR Decoder. This block increases the volume required to turn on a particular channel when that channel is on in the other direction. This reduces the possibility of the channel being on in both directions.

The Channel Assignment Update block keeps up with which channels have speech present and sends this information to the decoder. Since the encode and decode must stay in sync, the Channel Assignment Update block updates the status of a channel only after the information has been transmitted to the decoder.

The Channel Assignment Information (CAI) is critical to the proper interpretation of the data bits in the frame. Errors in this information will cause the loss of possibly all channels until the error is corrected. To insure the accurate transmission of this information, an error correcting system is used. The CAI Error Correcting Encoding block uses a correct one/detect two Hamming code [12] to generate four check bits for every four information bits. This system will correct all single errors and detect two errors out of each group of eight sent.

Bit Allocation computes the number of bits available for each channel (L) based upon the number of data channels

and the number of active voice channels. The # Data Channels Selector is a thumbwheel switch which can be dialed from 0 to 6. The Channel Assignment Update block provides the number of active voice channels to the Bit Allocation block The Signaling Extractor block picks off the last bit per word every sixth frame. These signaling bits are stored and sent to the multiplexer eight bits per frame.

The voice data also enters the Delay block which delays the data a fixed amount from 0 to 10 milliseconds. The purpose of this delay is to reduce front-end clipping of speech bursts. For example, if 10 ms are required to detect the voice burst and the data is delayed 4 ms, only the first 6 ms are actually clipped. This does, however, introduce an overall 4 ms delay.

The Code Converter block changes the incoming 8-bit μ255 companded words into 14-bit linear words. The μ255 code is a signed magnitude code logarithmically compressed to expand its dynamic range. The linear code is a 14-bit two's complement number. This conversion is necessary since the ADPCM algorithm is designed to operate on a linear code.

The ADPCM block applies this algorithm to the 14-bit incoming signal to produce an L-bit output which carries almost all the information to the decoder. L, which varies from eight to three as a function of loading, is supplied by the Bit Allocation block.

The Multiplexer block combines into each frame the signaling, CAI, voice data from the ADPCM block, and the dedicated data which bypasses the ADPCM algorithm. It adds the framing bit from T1A to complete the 193-bit frame. This output is converted to a standard bipolar T1 format for transmission over a regular 1.544 Mbit per second digital line.

The DDR Decoder receives the 1.544 Mbit per second bit stream into the Receive D block. Here the receive card converts the bit stream to unipolar and syncs on the framing bits. The Sync and Timing block generates the required timing and sync pulses for the remainder of the decode side.

The Bit Stream Decoder divides each frame into signaling, channel assignment, voice data, and dedicated data. The voice data bits are grouped into L-bit words and output at the proper time as an 8-bit word with zero fill. The channel assignment bits are output to the CAI Error Correcting block where most transmission errors are corrected. This corrected CAI is used by the Channel Assignment Update block. The Bit Allocation block uses the updated CAI and the number of data channels from the # Data Channels Selector to compute the number of bits per channel (L).

The ADPCM block performs the decode algorithm on the voice data words to generate a 14-bit linear code like the one at the transmitter. The Code Converter changes this back to an 8-bit μ255 companded code as used by the

T1 systems. Finally, the De-Multiplexer reconstructs the two T1 frames for output. The voice channels are deinterlaced, the signaling is added, and the framing bit is reproduced. The outputs of the DDR Decode are almost identical to the inputs to the DDR Encode.

3.2 ADPCM

The function of the ADPCM encoder is the conversion of 8-bit companded speech samples into an L-bit adaptive differential signal. This is accomplished using the functional diagram shown in Figure 3.4.

The input to the coder is an 8-bit word every 2.6 μsec. This companded speech sample is converted to a 14-bit two's complement linear number as required by the ADPCM algorithm. The first summing node subtracts the prediction from the sample to give the difference to the quantizing process. This process begins by limiting this difference to the current quantizing range which adapts as required. Next this range is mapped to an 8-bit word using multiplication. The L most significant bits (L varies from eight to three as a function of loading) are sent to the decoder. These L most significant bits are used at the encoder and decoder to reconstruct an 8-bit word which is then mapped back to a 15-bit difference. This reconstructed difference is added to the prediction to form the reconstructed sample. At the decoder this sample is converted back to an 8-bit companded word and is sent to a T1 receiver where it is decoded. Both the transmit and receive sides multiply this sample by .875 to form the prediction for the next sample. The L-bit output is also used to adapt the range. When the output approaches the limits of the quantizer the range number is increased. When the output falls within the

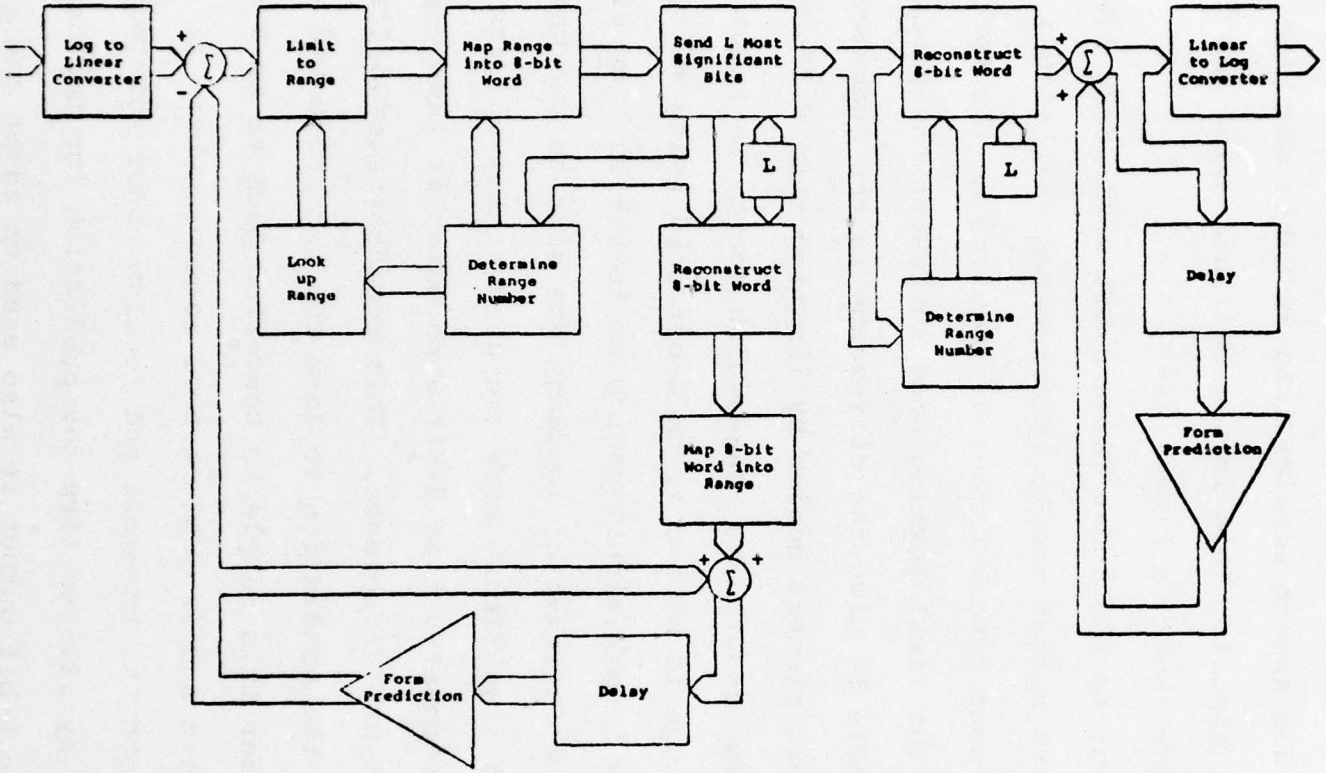


Figure 3.4 ADPCM block diagram

middle 50 percent of the range, the range number is decreased. The range is stored on a PROM which is addressed by the range number.

The coder must handle 48 channels simultaneously. This requires the processing of 48x8000 samples per second or one every 2.6 μsec. This requires the storage of the prediction and the range number for each channel. The only requirement made on the rest of the system is speed. This led to a mostly asynchronous design described in Figure 3.4. The implementation of the ADPCM coder required the development of some special numerical techniques. These will be developed before the actual algorithm is discussed.

A useful numbering system in digital work is the two's complement system. Positive numbers are represented by a zero followed by the number's magnitude, i.e., +5 is 0101. Negative numbers are represented by a one followed by the magnitude minus one, inverted. For example, -5 is 1 $\overline{\text{binary (5-1)}}$ or 1 $\overline{\text{binary 4}}$ or 1 $\overline{100}$ or 1011. A variation of this, called Modified Two's Complement (MTC) uses a one for the first bit of positive numbers and a zero for negative numbers. Table 3.1 shows a comparison of the techniques. MTC appears to be a non-signed binary number. This makes it an appropriate numbering system for use in mapping by multiplication as will be seen later.

Table 3.1 A comparison of decimal, binary, two's complement, and modified two's complement numbering systems

Decimal	Binary	Two's Complement	MTC
+3	+ 11	011	111
+2	+ 10	010	110
+1	+ 1	001	101
+0	+ 0	000	100
-1	- 1	111	011
-2	- 10	110	010
-3	- 11	101	001
-4	-100	100	000

An adaptive system requires an adapting quantizer whose range of operation varies with the statistics of the signal. The largest range must accommodate the largest possible signal and the smallest range should provide finely spaced levels for small amplitude signals. Sixty-three discrete ranges were chosen from ± 72 to ± 14848. Each range was assigned a range number from 0 to 62. The ranges are spaced uniformly on a logarithmic scale approximately two to the one-eighth ($2^{\frac{1}{8}}$) apart. For a list of these ranges, see Appendix A.

The first step in the ADPCM algorithm is the subtraction of the prediction from the current sample. The code converter outputs a 14-bit two's complement linear representation of the current sample. The prediction is subtracted from the current sample using two's complement arithmetic. This requires simply inverting the prediction and adding it and one to the current sample. The output is a 15-bit two's

complement number representing the difference to be quantized. Finally, the sign bit is inverted to change to the MTC numbering system.

This difference must now be limited to the current range of the quantizer. Figure 3.5 shows how this is done in MTC.

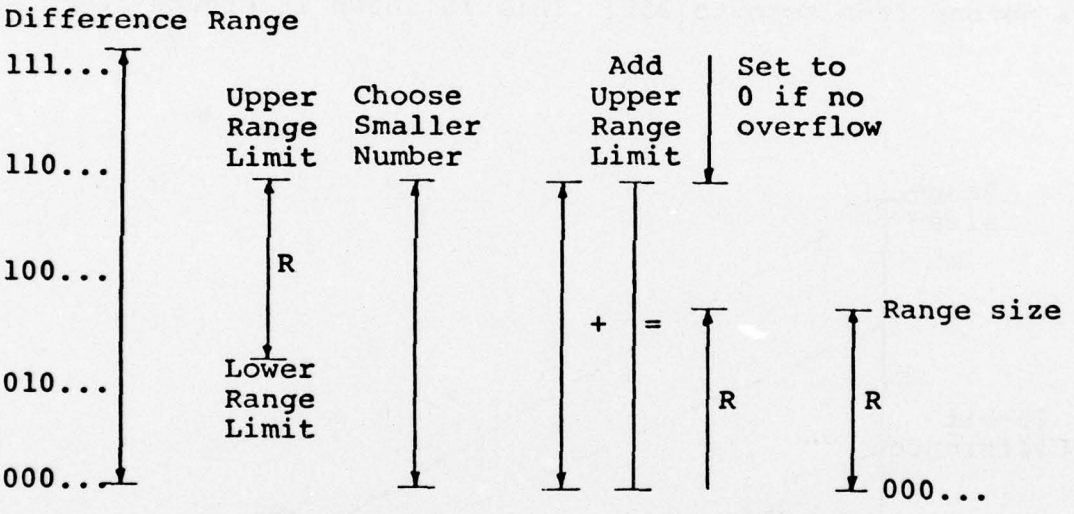


Figure 3.5 The limiting of the difference to the range using MTC

The difference is now expressed as a 15-bit MTC number. The range is centered about 100 000 000 000 000. The limits of the range are expressed as $40000_8 \pm \frac{\text{range}}{2}$. Thus, the smallest range limits are $40000_8 \pm 74_{10}$ or 40110_8 and 37670_8. First, the smaller of the difference and the upper range limit is chosen. This eliminates positive differences that are too large. Next, the upper range limit is added to the number chosen. This maps the range into the numbers

from zero to the range size. If no overflow is detected, the difference is set to zero. This process clamps the difference to the range and maps the range into the numbers starting at zero.

This 15-bit MTC number representing the difference must now be quantized to eight bits or one of 256 levels. Since the range of the difference is from zero to the range size, multiplication by $\frac{256}{\text{range size}}$ will map the range into a number from zero to 255. This is shown in Figure 3.6.

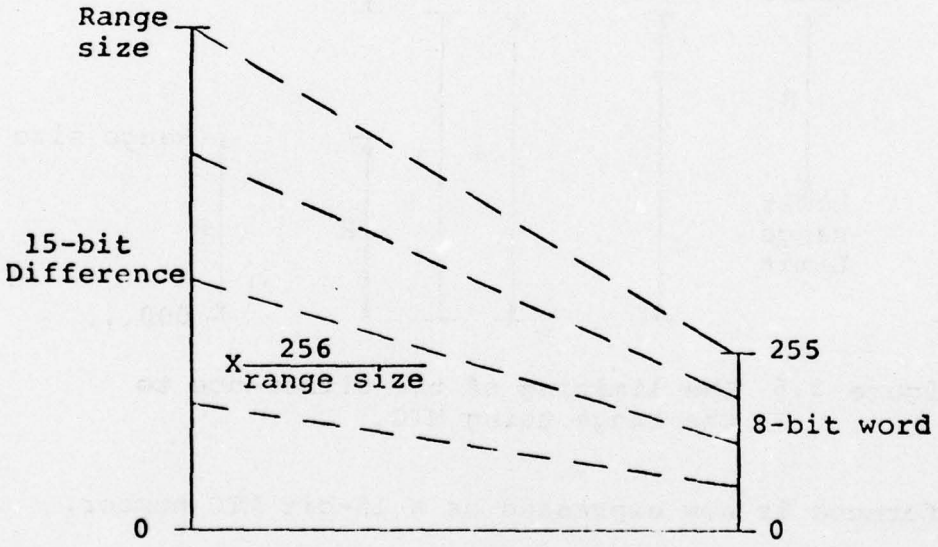


Figure 3.6 The mapping of the range into an 8-bit word using multiplication

This multiplying must be done 8000 samples x 48 channels per second or every 2.6 μsec. Time must also be allowed for other operations, leaving only about 500 nanoseconds for this multiplication. The simplest technique, shifting and

adding, is not fast enough for this. A combinational multiplier is expensive and would draw one amp at 5 volts. Using discrete chips would require 60 4-bit adders and a large number of shifters. The method selected requires 12 adders and 12 multiplexers. It involves selecting the range sizes such that neither they nor their reciprocals have more than four ones in their binary representation. Since

$$\frac{256}{\text{range size}} = 10000000_2 \times \frac{1}{\text{range size}},$$

the multiplier in this operation will never have more than four ones. Now only four 16-bit numbers need be added to effect the multiplication.

The next constraint on the range sizes is that they all be powers of two times the smallest eight range sizes. Since the range sizes were spaced about $2^{\frac{1}{8}}$ apart, this is already done. This constraint allows us to multiply by one of eight numbers and shift the result to obtain the proper power of two. The smallest eight range sizes and their reciprocals are listed in Table 3.2.

Clearly no number has more than four ones. Also, it happens in the reciprocals, that no number has a one in both the third and fifth place, or in both the fourth and sixth place. The first bit is always one, and no more than a single one occurs in digits two, seven and eight. This means that, when multiplying by $\frac{256}{\text{range size}}$, we can always add the multiplican and the multiplican shifted right by either two

Table 3.2 A list of the eight smallest ranges and their reciprocals

Range #	Range Size	Binary Range Size	Binary Reciprocal x 256
0	144	10010000	1.1100100
1	152	10011000	1.1011000
2	164	10100100	1.1001000
3	184	10111000	1.0110010
4	200	11001000	1.0100100
5	216	11011000	1.0011000
6	232	11101000	1.0001101
7	256	100000000	1.0000000
8	288	10010000	0.1110010
	Digit	12345678	12345678

or four places. Likewise, we can add the multiplican shifted right either three or five places and the multiplication shifted right by either one, six, or seven places. The sum of these subtotals is the difference times $\frac{256}{\text{range size}}$. The powers of two are taken care of by shifting the multiplican left an appropriate amount before multiplying. Figure 3.7 gives a functional diagram of this multiplication technique. Control of the shifting operations is derived from the range numbers using combinational logic.

The 8-bit output word of the multiplier is next reduced to L bits. L is the number of bits permitted per sample. It varies from eight under light loading to three under the most severe loading. All but the L most significant bits are changed to zero. This new 8-bit word is sent to the multiplexer which picks off the L most significant bits and sends them to the decoder.

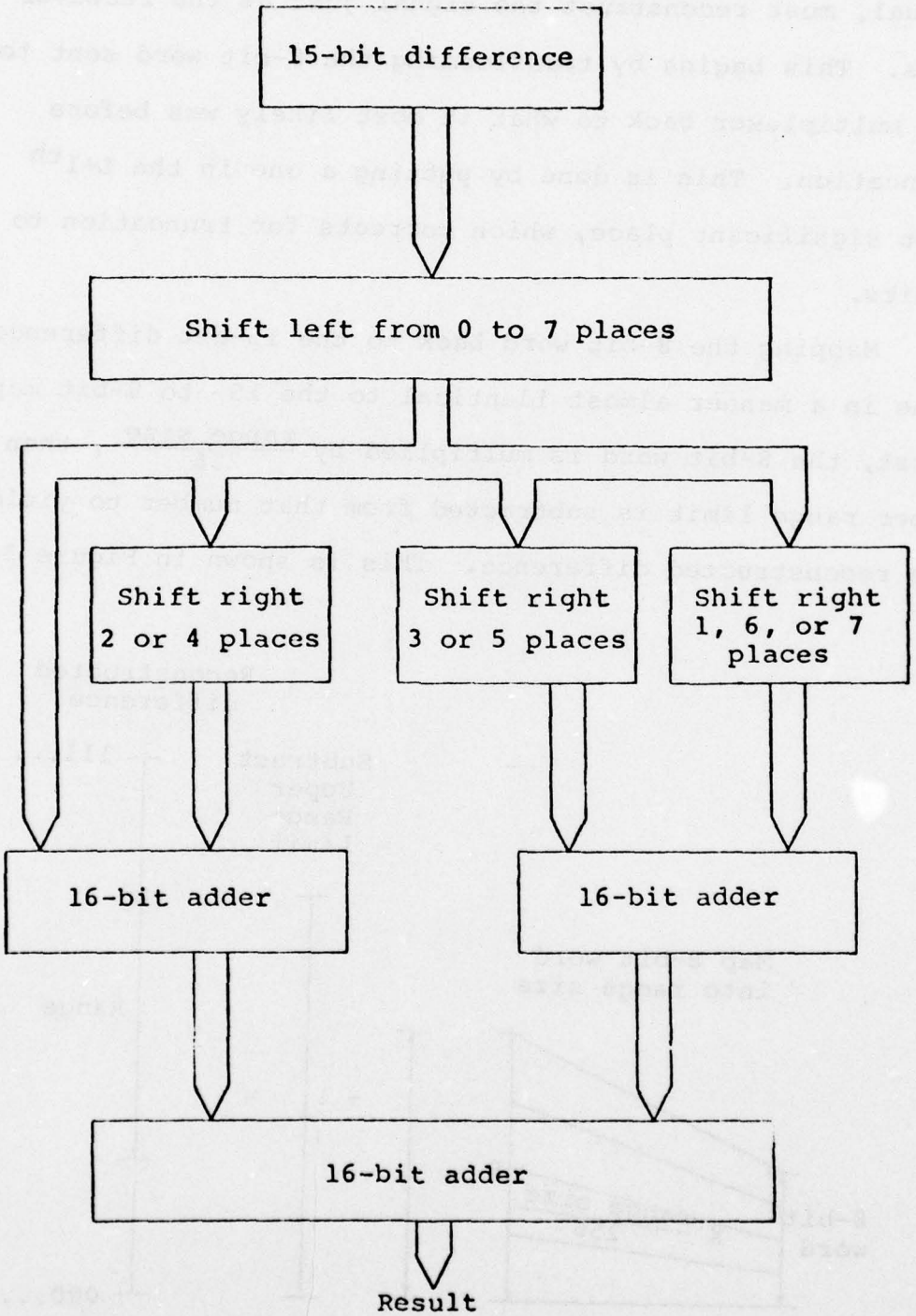


Figure 3.7 Block diagram of a special purpose multiplication technique

The encoder, in order to send the proper difference signal, must reconstruct the signal just as the receiver does. This begins by transforming the 8-bit word sent to the multiplexer back to what it most likely was before truncation. This is done by putting a one in the $L+1^{th}$ most significant place, which corrects for truncation to L bits.

Mapping the 8-bit word back to the 15-bit difference is done in a manner almost identical to the 15- to 8-bit mapping. First, the 8-bit word is multiplied by $\frac{\text{range size}}{256}$, then the upper range limit is subtracted from that number to yield the reconstructed difference. This is shown in Figure 3.8.

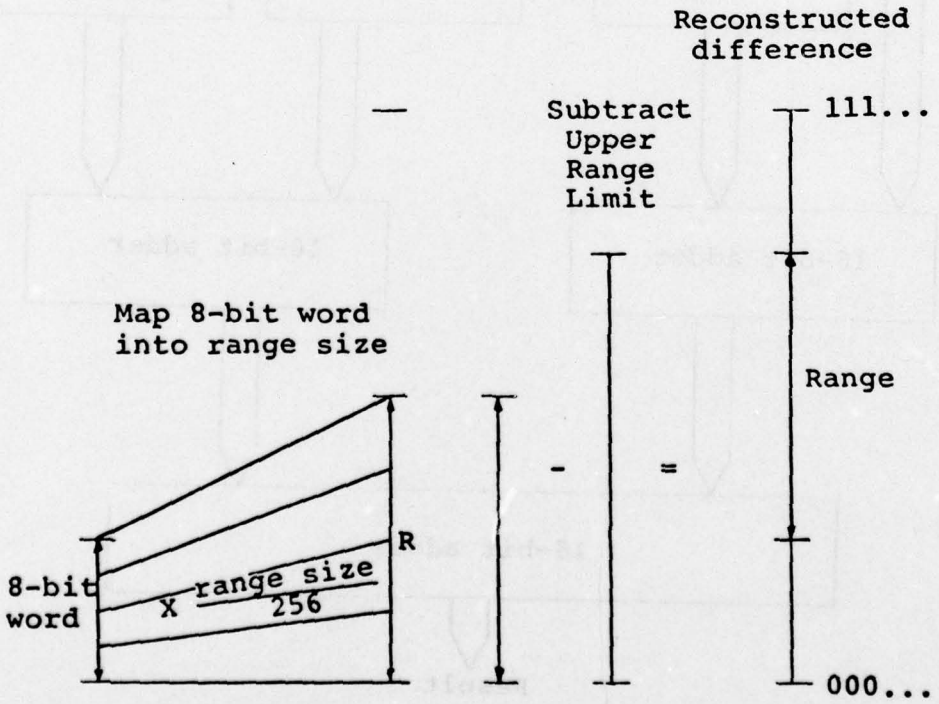


Figure 3.8 The mapping of an 8-bit word into the range using multiplication

The reconstructed difference is now added to the prediction to get the reconstructed sample. This is the same value the decoder will have if no transmission errors occur. This value is stored in a memory while the other 47 channels are being processed. It is then recalled and multiplied by .875 to form the next prediction. This completes the loop.

The range adaptation procedure is examined next. The encoder and decoder must use the same range size to properly interpret the signals. The only information shared by the encoder and decoder is the L-bit output. To eliminate the requirement to transmit the range sizes, one must base the range adaptation on the current L-bit output.

The range adaptation strategy is the one recommended by Jayant [8] modified slightly. He changes his step sizes by multiplying by some factor based on the magnitude of the five most significant bits. Since DDR range sizes are powers of $2^{\frac{1}{8}}$ times the smallest range size, it is restricted to multiplying by powers of $2^{\frac{1}{8}}$. Also, the multipliers are based upon the L most significant bits. Finally, since the range sizes are numbered, changes are made in the range size by adding to or subtracting from the range number. Figure 3.9 shows the similarity of the two methods. A list of the entire adaptation strategy appears in Appendix B.

The implementation of the algorithm simply requires that the range number be updated after the L-bit output. The output addresses a ROM which supplies the range number adjustment. The new range number is stored in a memory.

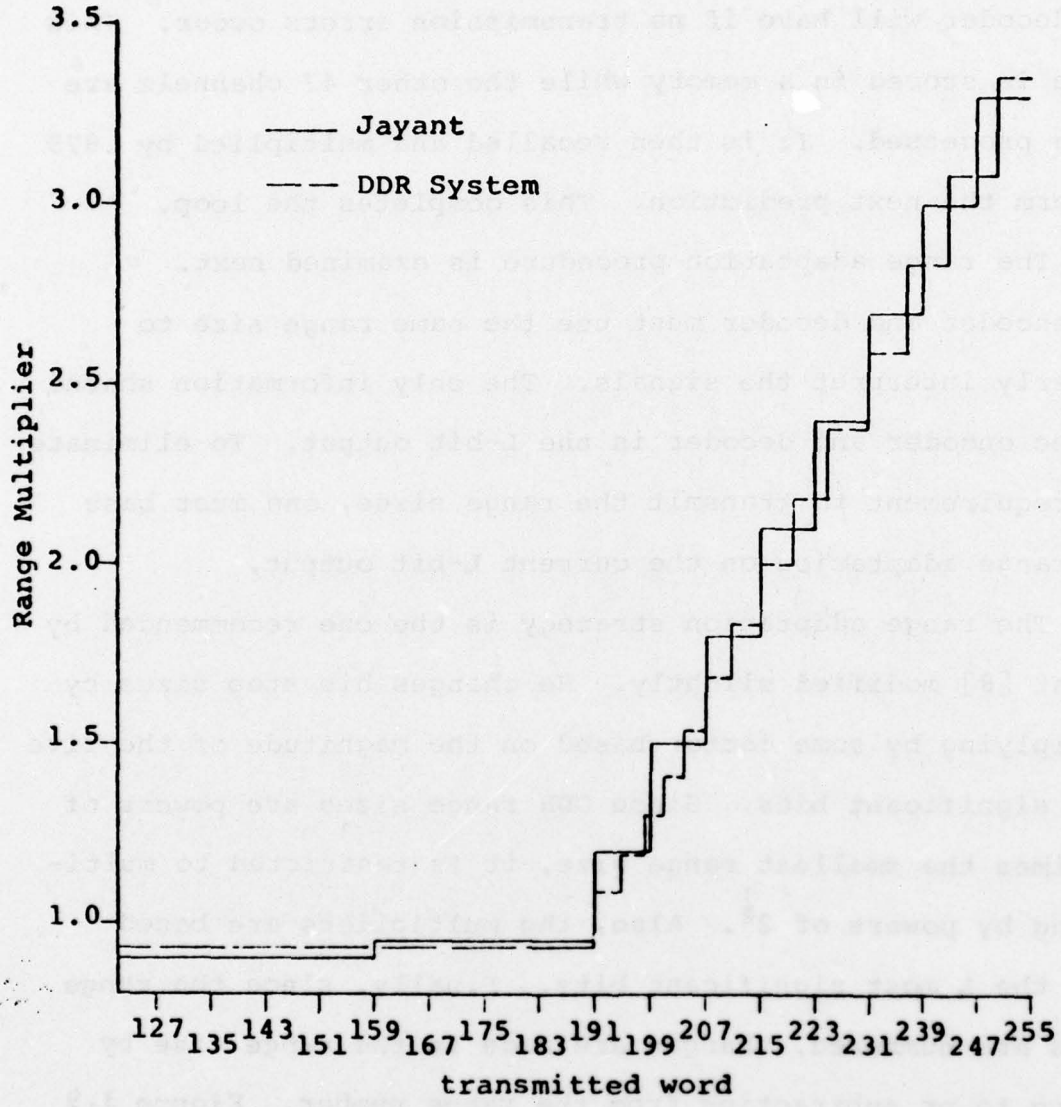


Figure 3.9 Comparison of Jayant's Adaptation Strategy and DDR Adaptation Strategy

During the processing of the next sample, the new range number addresses a ROM which outputs the new upper range limit. The receiver works in the same manner.

3.3 Expected DDR Performance

The expected performance of the DDR system based upon computer simulations [1] and design considerations is summarized in this section. The performance of the prototype system has not yet been experimentally determined but it is expected to follow the expected performance. The criteria for evaluating the performance are the signal-to-noise (S/N) ratio, bandwidth, ability to handle modems, overall delay, speech clipping, idle channel noise, and susceptibility to line errors. The performance with respect to each criterion is discussed below.

The most commonly used indicator of performance for this type of system is the S/N ratio. The T1 system achieves a S/N ratio of about 36 dB. Computer simulations show that under light loading the DDR system achieves a S/N ratio of 34 dB [1]. The noise power from two independent sources can be added in the following manner to obtain an overall S/N ratio:

$$S/N_{overall} = -10 \log_{10} \left(10^{-\frac{S/N_1}{10}} + 10^{-\frac{S/N_2}{10}}\right).$$

Assuming independent noise sources, the overall S/N ratio at the receiver would be 31.9 dB, or a loss of 4.1 dB caused by

the DDR system. The most severe loading (six dedicated data channels and 42 off-hook voice channels) reduces the S/N ratio at the receiver by only 1 dB. This is shown in Figure 3.10.

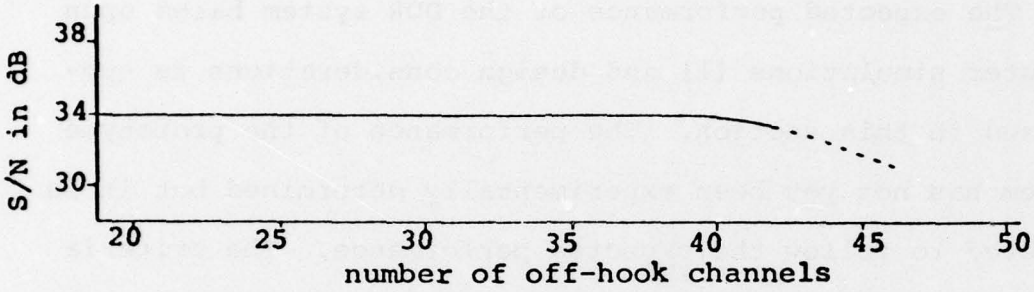


Figure 3.10 ADPCM/TASI performance—most severe loading [1]

These S/N ratios are long-term (2 to 4 second) averages and are not necessarily an accurate reflection of subjective quality. For instance, a long period at 36 dB followed by a short burst of 25 dB may average 34 dB but sound worse than a constant 34 dB. Although the short-term performance of the system is not well defined, limited simulations have not shown this to be a problem.

The bandwidth transmitted by the T1 system is limited by the sampling rate of 8000 samples per second to 4000 Hz. Since the DDR also transmits 8000 samples per second, the bandwidth will be unaffected. However, the performance of the ADPCM algorithm is frequency dependent. Since only the difference is sent, low frequency signals are transmitted more accurately than high frequency signals. This is acceptable for speech because most of the energy is in the lower frequencies and the higher frequencies are already noiselike.

However, modems, which use the center frequencies of the voice channel, will probably operate poorly with the ADPCM algorithm. For this reason the DDR is designed to accept up to six channels of dedicated data which bypass the ADPCM algorithm. This requires the manual setting of a switch at each end of the system. The format of this data is unrestricted. Each data channel may contain 64,000 bits/sec.

The use of TASI in this system requires the examination of the performance with respect to speech clipping, delay, and idle channel noise. Speech clipping is the failure to transmit the initial portion of a speech burst. Table 3.1 shows the effect of various amounts of clipping.

Table 3.1 Effect of various clip lengths [10]

Length of Clip	Effect
Less than 15 ms	Imperceptible
Less than 30 ms	Does not affect the articulation of fricatives
Less than 40 ms	Does not affect the articulation of semivowels
More than 50 ms	Significantly reduces articulation

The length of the speech clipping is a function of three factors: voice switch delay, channel assignment delay, and in-line delay. These are shown in Figure 3.11.

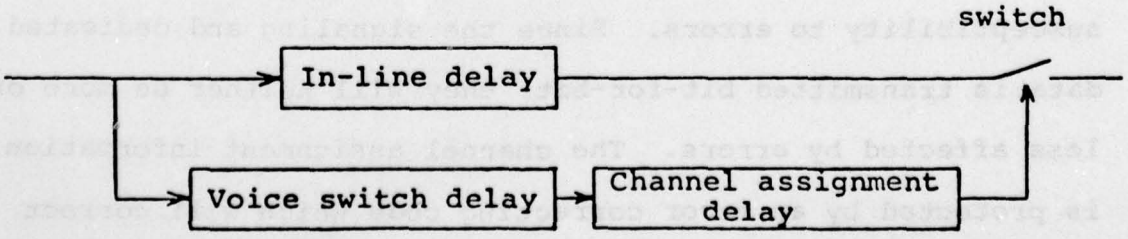


Figure 3.11 Source of speech clips in the DDR system

The voice switch delay is the length of time it takes the voice switch to detect speech. This is always less than 20 ms [10]. The channel assignment delay is the length of time it takes to transmit to the receiver the fact that a particular channel is active. This takes between .5 and 3 ms. These two delays are cumulative. The in-line delay simply delays the speech a selectable amount from 0 to 10 ms. This reduces the speech clipping by the amount of the delay, but introduces overall delay into the system. This delay can increase the severity of echos in the system. A compromise of about 4.5 ms should render the speech clipping imperceptible.

Since the DDR does not transmit when no voice is present on a channel, there will essentially be no idle channel noise. This is a problem only when there is background noise on the input signal. The turning on and off of this noise with the speech is noticeable and somewhat objectionable. This problem can be solved by adding noise at the receiver when no speech is present, but this is not done in the current DDR design.

The effect of transmission errors is considered next. The removal of redundancy in a signal tends to increase its susceptibility to errors. Since the signaling and dedicated data is transmitted bit-for-bit, they will neither be more or less affected by errors. The channel assignment information is protected by an error correcting code which will correct more than 99 percent of the errors. Since 8-bit ADPCM is

subjectively less susceptible to errors than 8-bit PCM [13] the DDR will be less affected by errors under light loading. However, as the bits per sample drops below eight, the voice data will become more susceptible to errors than PCM.

The performance of the DDR is expected to closely approach the design objective of tranparency. The actual performance of the system as built will be determined in the near future.

4. COMPONENT DESIGN

4.1 The DDR Shelf

The DDR system was built in a standard T1 carrier shelf so it has the same dimensions as an ITT 324 T1 carrier terminal. It occupies 14" of vertical space on a standard 19" rack. Existing T1 common equipment cards were used where ever possible. The system was constructed primarily from TTL ICs – standard TTL and low power Shottky TTL was used. The system described here is a prototype model built to determine the feasibility of such a system. Future models of this system could occupy much less space and utilize microprocessor technology. In what follows each of the blocks in Figure 3.3 is described. Many of the schematic diagrams in this chapter may be difficult to use because they have been reduced to standard 8½x11" size. A complete set of full-sized schematic diagrams is available from J.B. O'Neal, Jr., NC State University, PO Box 5275, Raleigh, NC. 27650. A complete set of drawings will be furnished on request to anyone demonstrating an official need for their use.

4.2 ADPCM Encoder and Decoder

In the DDR system ADPCM must be performed on all 48 channels. Digital voiceband data channels are processed but their ADPCM output is ignored by the multiplexer. The following paragraphs give a description of the hardware that implements DPCM.

Figure 4.1 shows how the hardware code conversion and ADPCM might be implemented in the DDR system. The DPCM operation at the transmitter consists of the two summing nodes, a 0.875 multiplier, and a unit delay memory. Two's complement binary arithmetic is used so that the additions and subtractions at the summing nodes can be implemented with available 4-bit adders. Since 0.875 = 1 - (1/8), the multiplication can be performed by hardwiring a shift right operation into a bank of adders. For example, consider the two's complement number

011011.

A shift right operation, with sign extension, gives

000011.

Subtraction from the original number in two's complement is done as follows:

```
  011011    (original)
+ 111100    (right shifted and inverted original)

1 010111    (neglect carry)
+      1    (for two's complement subtraction)

  011000    (answer) .
```

Since 011011_2 is 27_{10} and $(.875)(27) \simeq 24_{10} = 011000_2$, the answer checks. The unit delay can be easily implemented using a properly addressed RAM.

DPCM at the receiver consists of a summing node, a multiplier, unit delay memory and overflow circuitry. The summing node is implemented using the same type of two's complement addition as

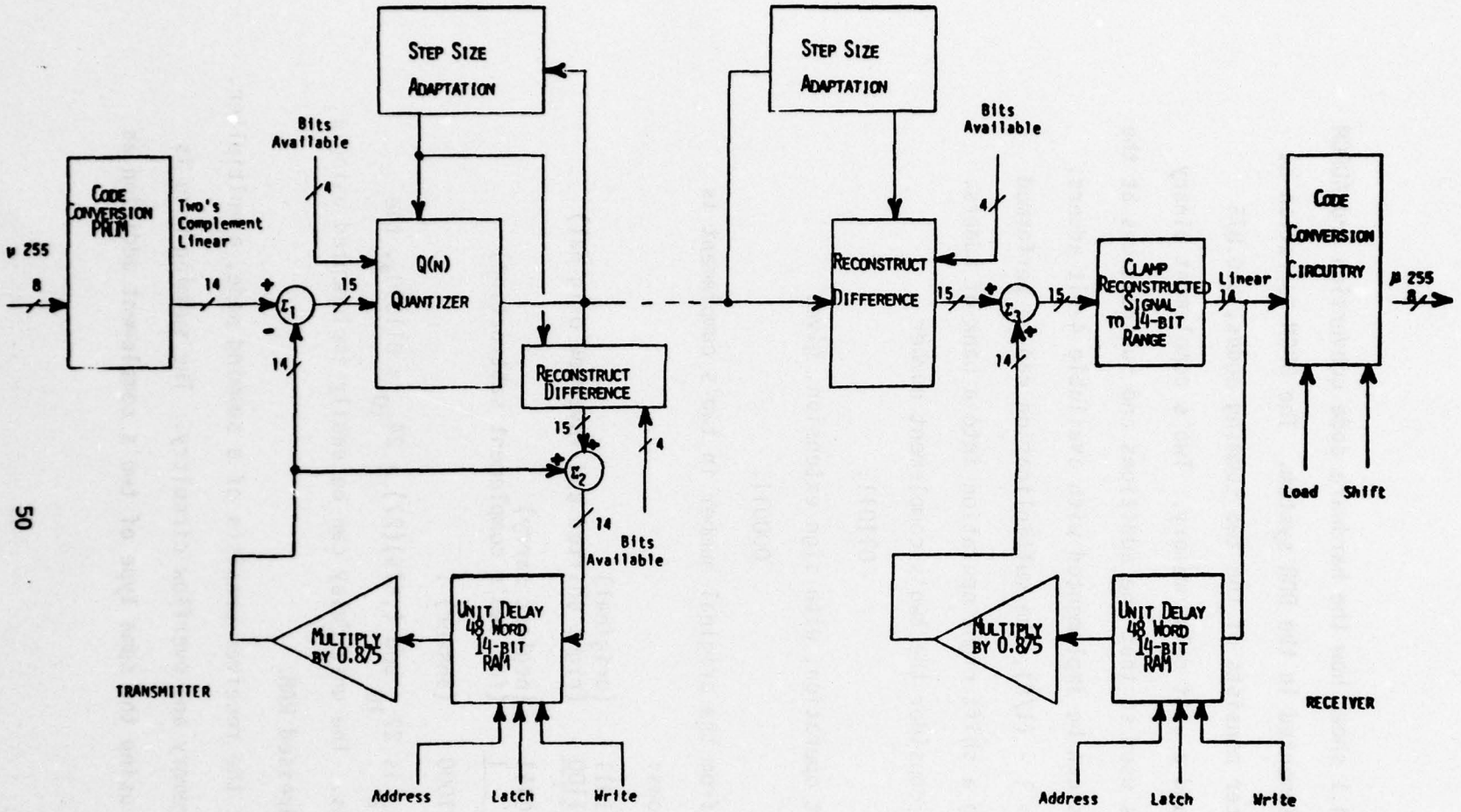


Figure 4.1 Hardware Oriented Block Diagram of the ADPCM Encoder and Decoder

the transmitter. Multiplication and delay hardware is identical to that of the transmitter. The need for overflow circuitry arises from the fact that errors due to a noisy channel may cause abs (d_i) to be excessively large. The overflow circuitry clamps s_i at the receiver to the top or bottom of its range.

Figure 4.2 gives a logic diagram of the code conversion PROM and the DPCM circuitry at the transmitter. PROMS d1 and d3 output a 14-bit linearly coded two's complement speech sample. This binary number is subtracted from the prediction, Σ_1 of Figure 4.1, by adders d9, d10, d11, and d12. Since the quantizer at the encoder and decoder uses modified two's complement, the most significant bit of binary words received from or sent to the quantizer must be inverted. The output of these adders is processed by the adaptive quantizer. Adders d49, d50, d51, and d52 add the reconstructed difference, RC, back to the prediction in order for the next prediction to be made. Since the prediction, XP, is stored in inverted form, inverters d53, d54, and parts of d55 are used to re-invert the signal for addition to RC. Adders d33, d34, d35, and d36 combined with inverters d55, d41, and d42 perform the 0.875 multiplication discussed above. The output of these adders is the next prediction. Latches d25, d28, and d29 prevent race conditions by latching the next prediction before the RAM write pulse occurs. RAMS d17 and d19 output the prediction during the first seven-eighths of a channel's address time and store the next prediction during the last one-eighth of a channel's addressing time. The 48 addresses, 0-47, are

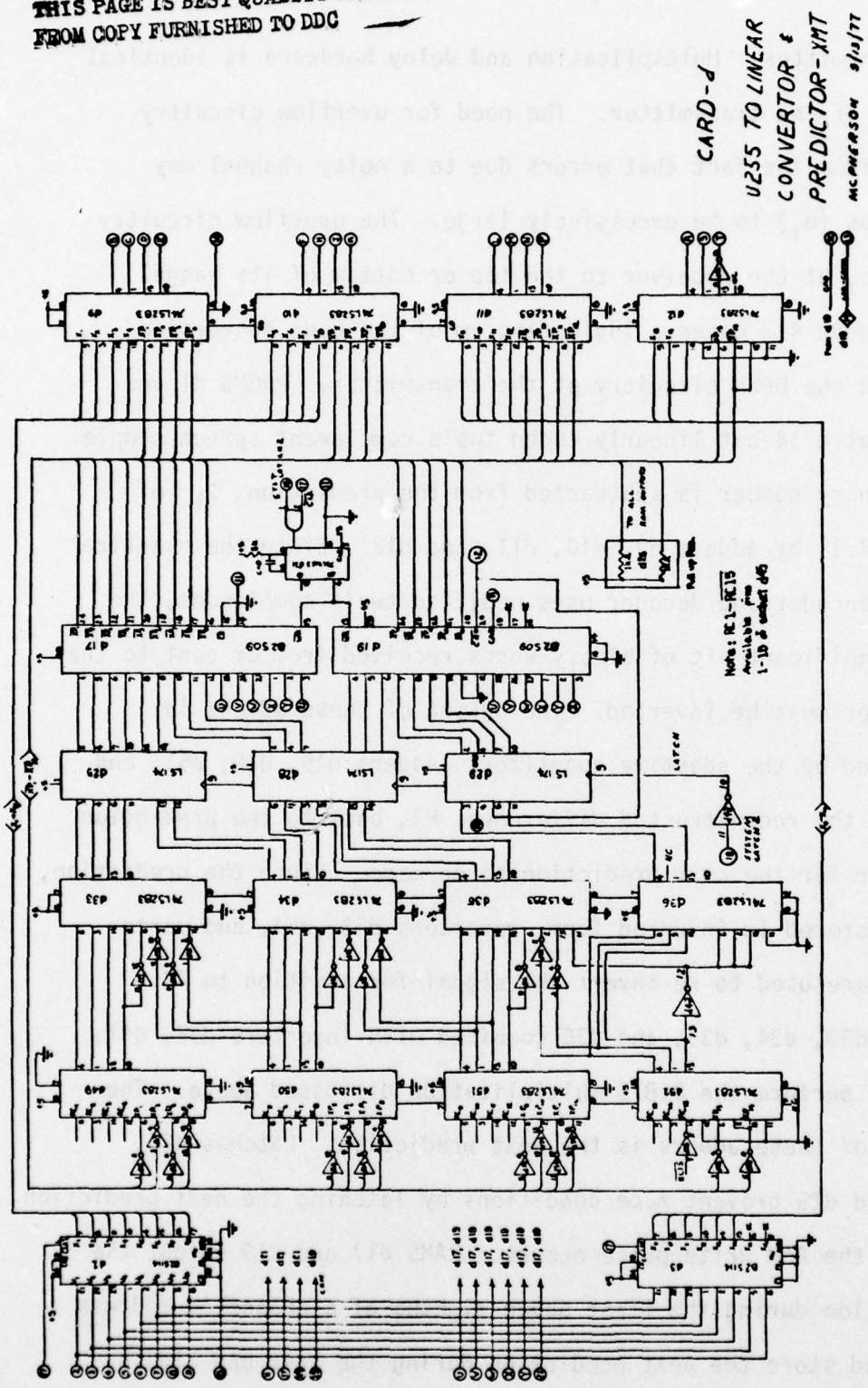


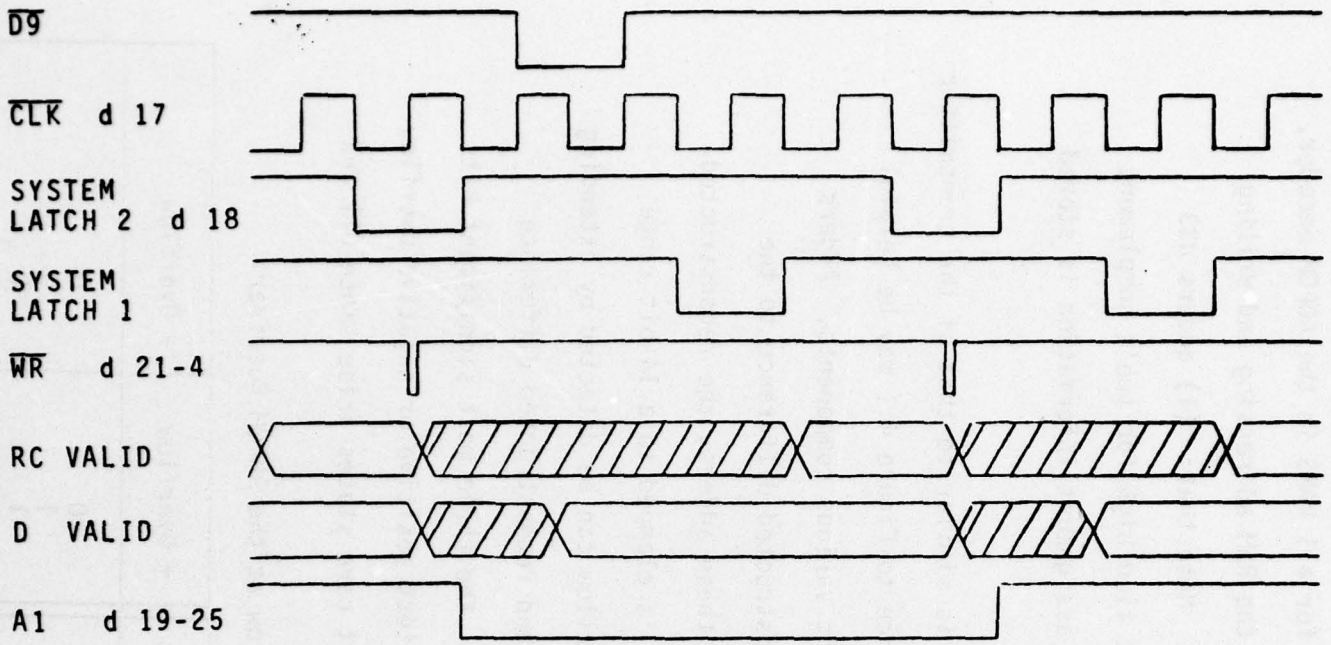
Figure 4.2 Logic Diagram of
Code Conversion and DPCM Circuitry
for the DDR Transmitter

taken from the voice switch timing as is System Latch 2. One-shot d21 provides a write pulse for all RAMS in the ADPCM encoder. Figure 4.3 gives the timing for the RAM addressing and writing in addition to data valid times. Note that: (1) adders d33 and d9 have their carry-in input tied high for two's complement subtraction and (2) the channel assignment information is stored in RAM d19.

The DPCM decoding hardware is similar to that of the transmitter as shown in Figure 4.4. Reference to Figure 4.1 may be helpful in realizing the functions of the various components. Adders w1, w2, w3, and w4 add the reconstructed difference to the prediction, PRC. The output of these adders, the reconstructed linearly encoded speech sample, is clamped to a 14-bit range when overflow is detected. Overflow can be detected by extending the sign bit on the prediction and reconstructed difference so that 16-bit words are added. The three most significant bits of the result can be used to detect positive or negative overflow as given in Table 4.1 The don't care states arise since 011 and

Table 4.1 Overflow at the ADPCM Receiver

Bit 16	Bit 15	Bit 14	+ Overflow	- Overflow
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	d	0
1	0	0	0	d
1	0	1	0	1
1	1	0	0	1
1	1	1	0	0



Timing for code conversion and DPCM
Transmitter Card-d

Figure 4.3 Timing for Transmitter Code Conversion and DPCM

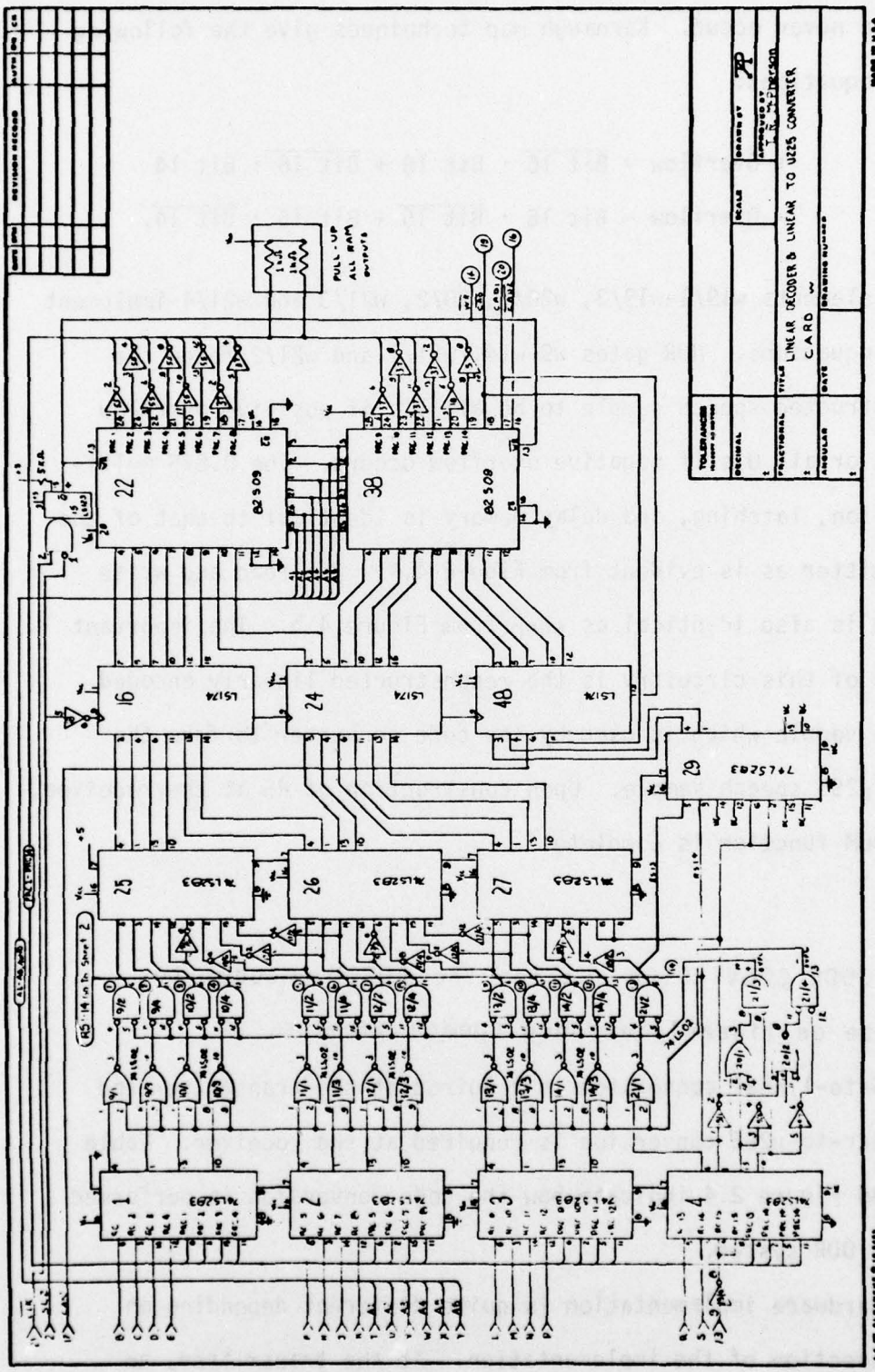


Figure 4.4 Logic Diagram
for Receiver DPCM circuitry
— Card W

100 can never occur. Karnaugh map techniques give the following logic equations:

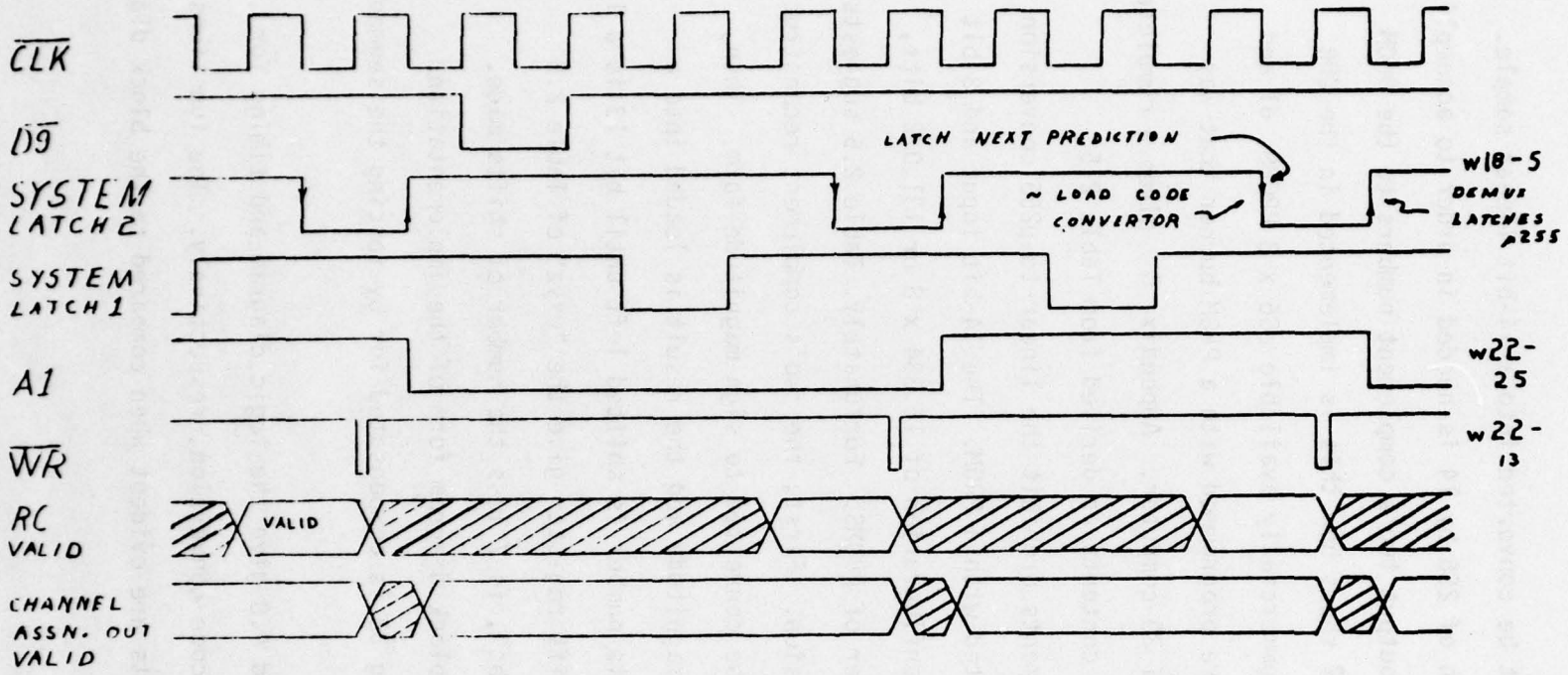
$$+ \text{Overflow} = \overline{\text{Bit 16}} \cdot \text{Bit 15} + \overline{\text{Bit 16}} \cdot \text{Bit 14}$$

$$- \text{Overflow} = \text{Bit 16} \cdot \overline{\text{Bit 15}} + \text{Bit 16} \cdot \overline{\text{Bit 14}}.$$

Logic elements w19/1-w19/3, w20/1,w20/2, w21/3 and w21/4 implement these equations. NOR gates w9-w14, w21/1 and w21/2 force the reconstructed speech sample to be all 1's if positive overflow occurs or all 0's if negative overflow occurs. The 0.875 multiplication, latching, and delay memory is identical to that of the transmitter as is evident from Figure 4.4. The read and write timing is also identical as seen from Figure 4.5. The important output of this circuitry is the reconstructed linearly encoded speech sample which is used by the code converter to form the 8-bit μ255 speech sample. Upon construction of RS at the receiver, the DPCM function is complete.

CODE CONVERSION enables the ADPCM encoder to operate on linearly encoded speech samples. A μ255-to-linear conversion is required at the transmitter and a linear-to-μ255 conversion is required at the receiver. Table 2.5 and Figure 2.4 indicate how the code conversion is performed in the DDR system.

Hardware implementation is quite different depending on the direction of the implementation. At the transmitter, an



DPCM DECODER TIMING

CARD w

Figure 4.5 Receiver DPCM Decoder Timing

8-bit μ255 word must be converted into a 14-bit linear sample. A PROM configuration of 256 by 14 is needed in order to accomplish the conversion and output two's complement numbers to the DPCM encoder. Figure 4.2 shows how this is implemented in the DDR system, using two commercially available 256 x 8 PROMS, d1 and d3. These PROMS were programmed with a PROM burner that was interfaced to an AGT 30 computer. Appendix C gives a complete listing of the PROM contents as derived from Table 2.5.

Memory requirements prohibit the linear-to-μ255 conversion from being implemented with a PROM. The 14-bit input and 8-bit output requires a configuration of 16,384 x 8 or 131,072 bits, an impractical number of PROMS. Fortunately, Table 2.5 suggests a method for conversion. First, the two's complement reconstructed speech sample must be converted to sign-magnitude form. Then, 33 is added to the magnitude and the result is loaded into a shift register. This number is shifted left until bit 13 is a 1. Bits 9-12 of the shift register give the "wxyz" of Table 2.5 and the segment, "abc", is 7 less the number of shifts made. Figure 4.6 gives a block diagram form of the implementation. Overflow from adding 33 is compensated for by forcing the segment number to 111.

Figures 4.7 and 4.8 give the logic diagram and timing for the linear-to-μ255 code conversion, respectively. The functions of the logic elements are evident when compared to the block diagram.

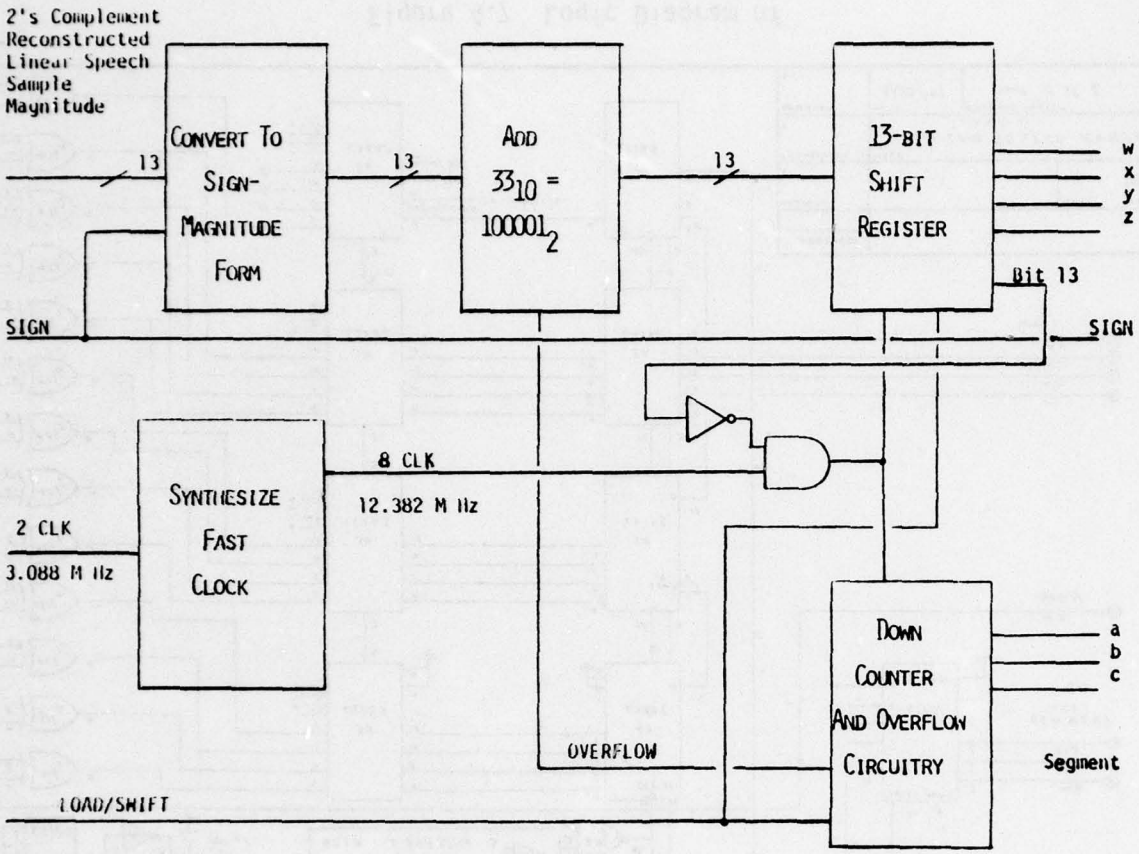


Figure 4.6 Linear-to-Logarithmic Code Conversion at the DDR Receiver

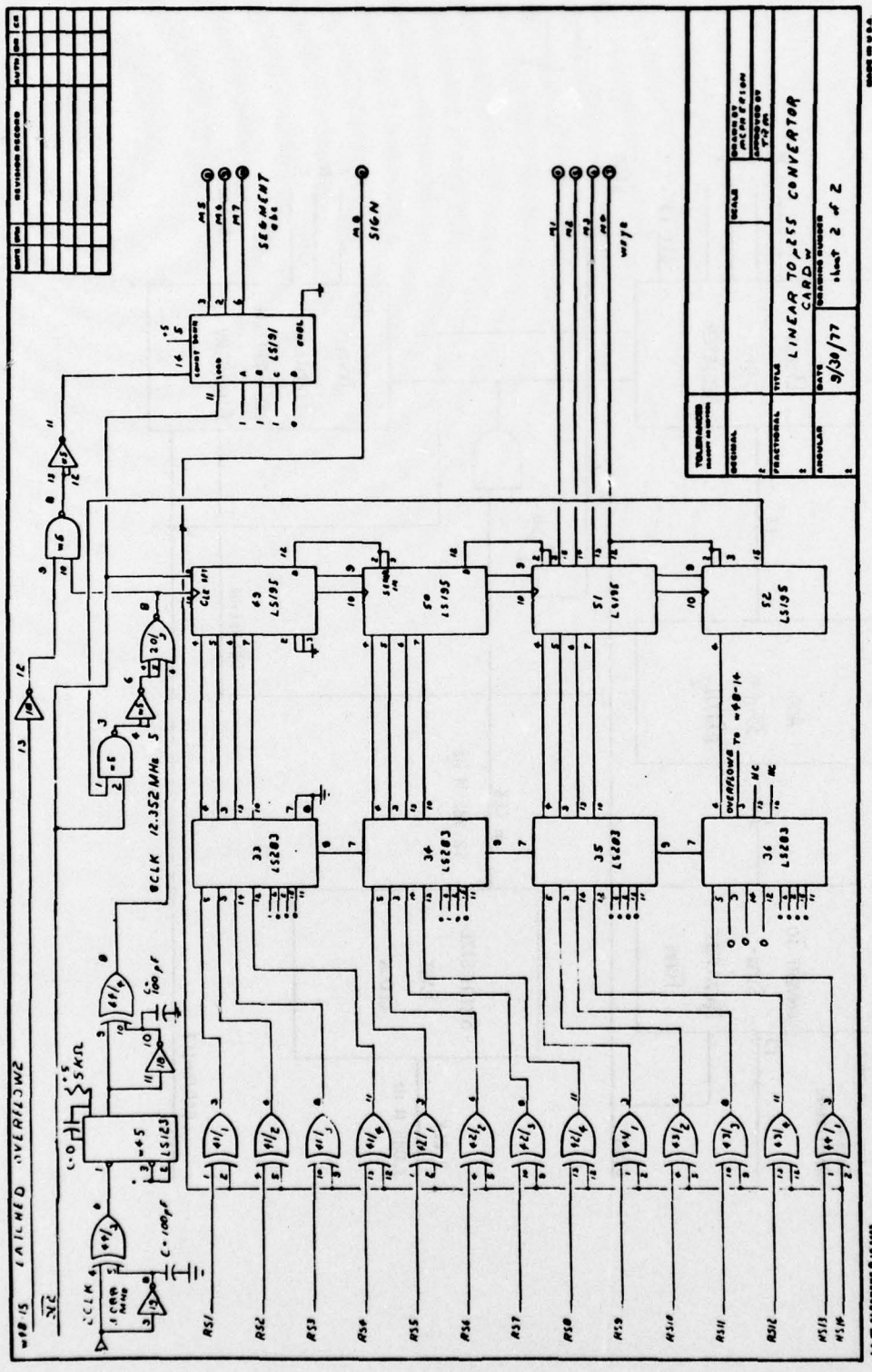
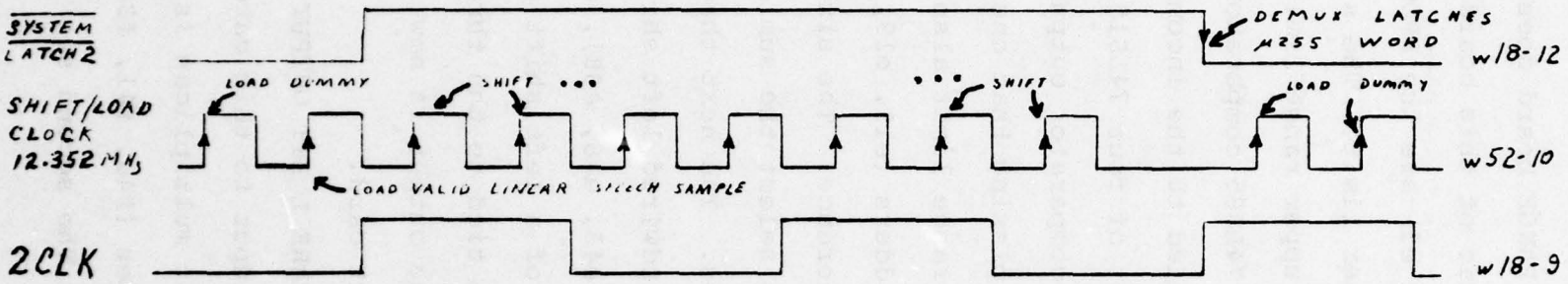


Figure 4.7 Logic Diagram of Linear-to-Logarithmic Code Conversion at the DDR Receiver



NOTE: THE NUMBER OF SHIFTS MAY VARY FROM 0 TO 7 DEPENDING ON THE MAGNITUDE OF THE LINEAR SPEECH SAMPLE

CODE CONVERSION TIMING
CARD-w

Figure 4.8 Timing for the Linear-to-Logarithmic Code Conversion at the DDR Receiver

The CLAMP DIFFERENCE TO RANGE board does just that. Figures 4.9 and 4.10 show the schematic of this board. Starting at the left, the two 825114s (e1 and e3) are the PROMs that convert the range number to the range upper limit. The six inputs 9 through 14 are the range number. The upper range limit is output to the A inputs of the bank of four 74LS85 comparators (e9, e17, e25, and e33). The B inputs are tied to the incoming 15-bit difference are also the inputs to the set of four 74LS157 multiplexers (e10, e18, e26, and e34). The A>B comparator output selects the smaller of these two numbers by multiplexing that one to the output of the multiplexers. The upper range limit also goes to the A inputs to the bank of four 74LS283 adders (e11, e19, e27, and e35) which adds this to the limited difference. The sixteenth bit is an overflow bit which is used to select the sum if overflow occurs and zero if no overflow occurs. The next three banks of multiplexers are used to select hardwired left shifts of 4 or 0 (e13, e21, e29, e37), 2 or 0 (e44, e45, e46, e38), and 1 or 0 (e43, e42, e41). This allows selection of a left shift of from zero to seven places. The select lines are tied to the three most significant bits of the range number. The output is now 12 bits and is ready for multiplication on the next card.

The DIVIDE BY RANGE-PREPARE L BIT OUTPUT card is shown in Figures 4.11 and 4.12. The input to this card is the output of the previous card. The 12-bit multiplican is input to three sets of multiplexers. The first set (f49, f41, f33) selects a hardwired right shift of 3 or 5. The second set (f50, f42, f34)

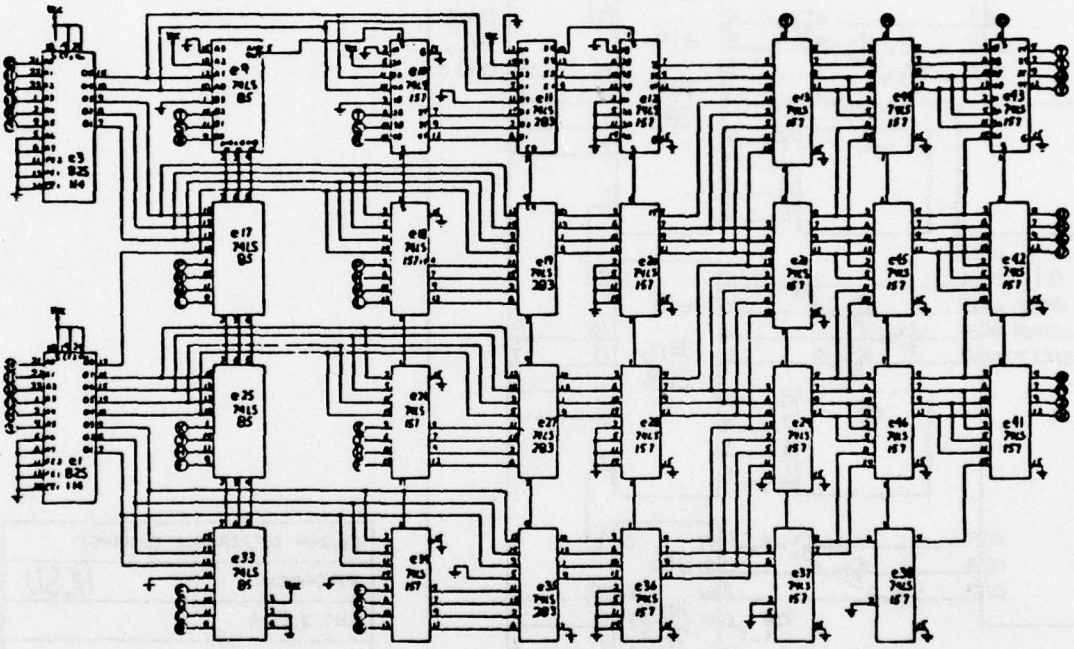


Figure 4.9 Clamp difference to range

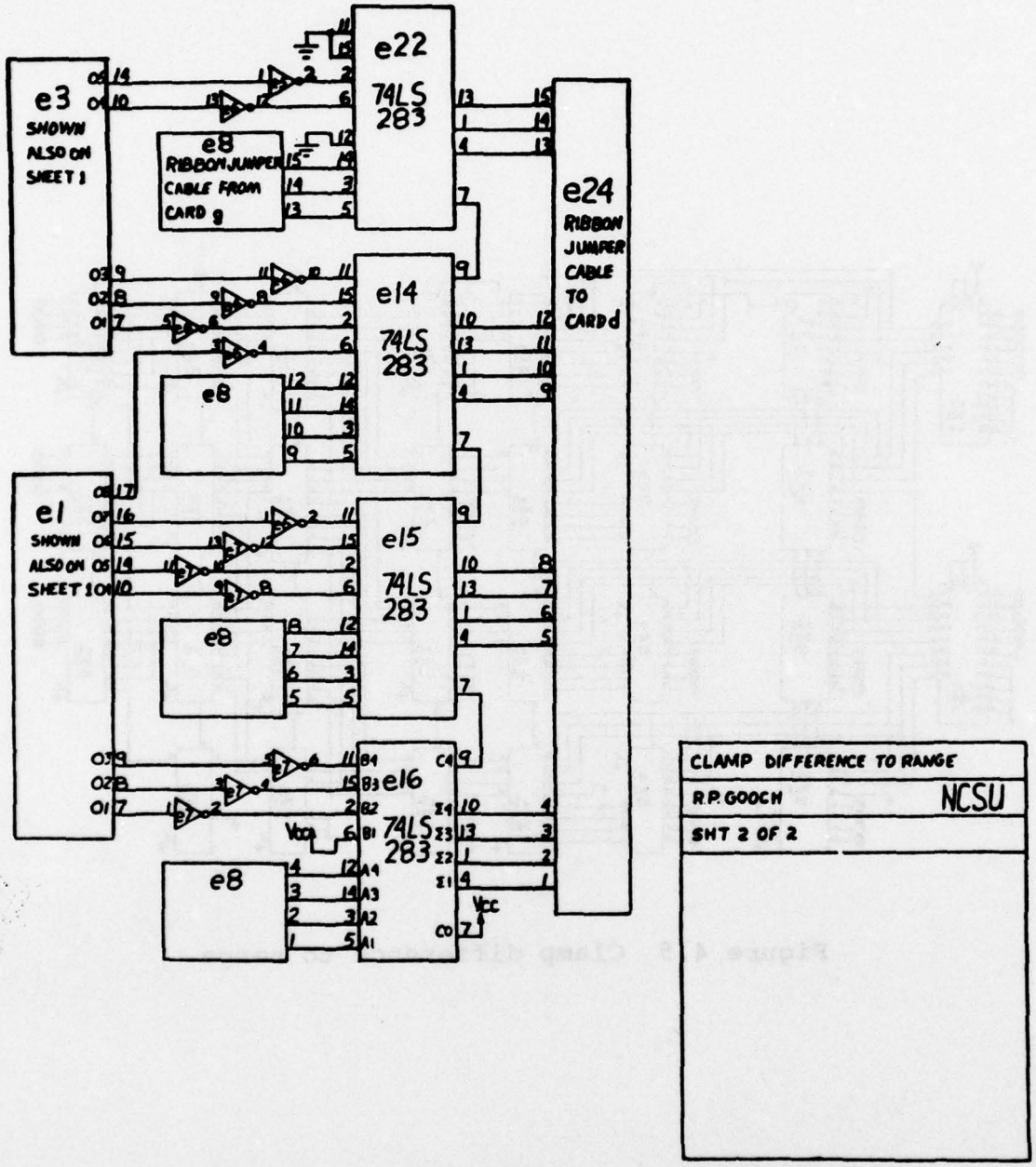


Figure 4.10 Clamp difference to range

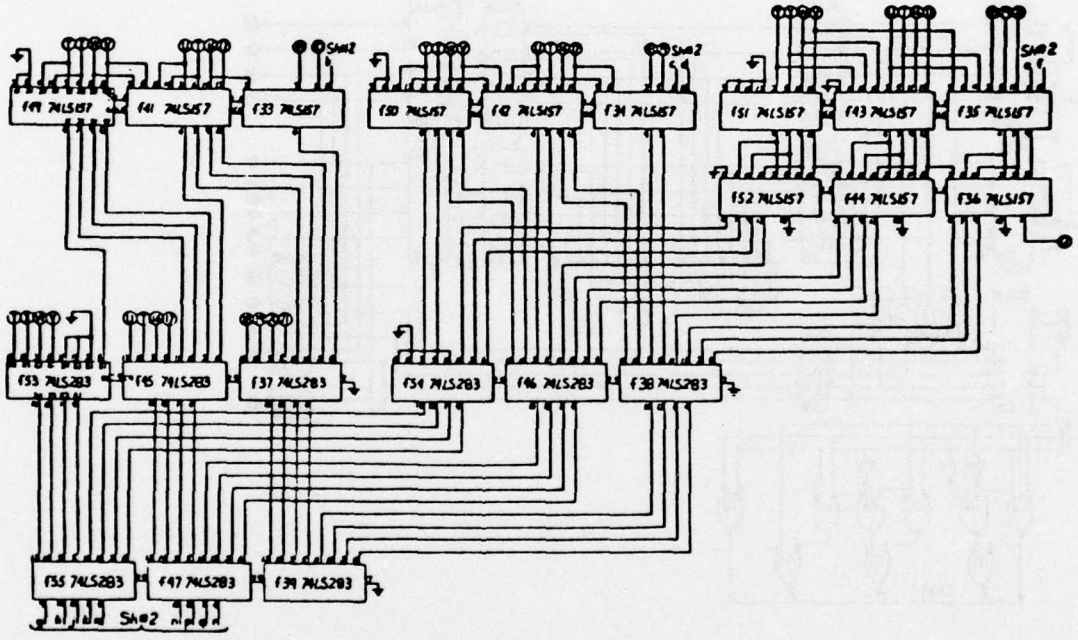


Figure 4.11 Divide by range - prepare L bit output

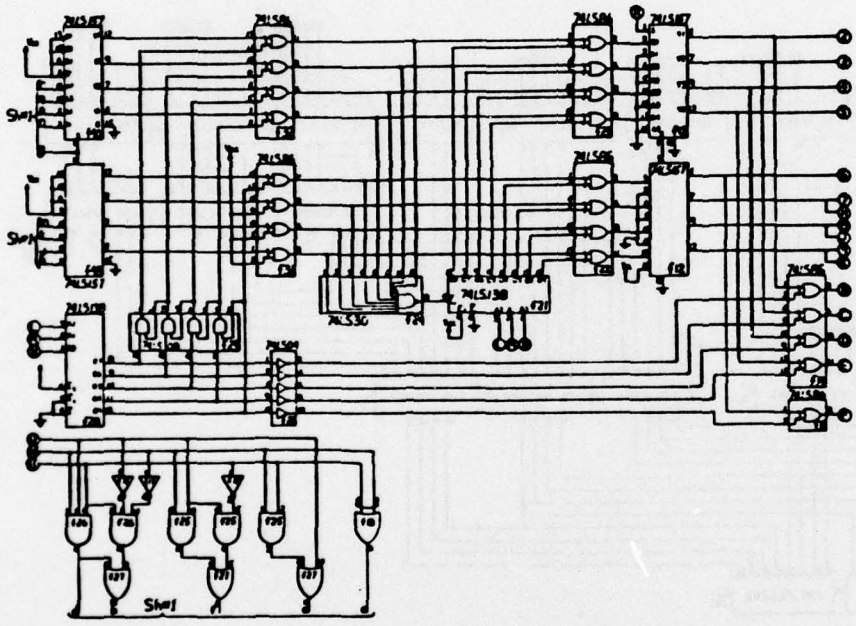


Figure 4.12 Divide by range - prepare L bit output

selects a 2- or 4-place shift, and the third set (f51, f43, f35) selects a 1- or 6-place shift. The third set is followed by a 0 or 1 shift (f52, f44, f36) to make it an overall 1-, 6-, or 7-place shifter. The output of each shifter bank can be set to zero as required by making the pin 15's high. All shifting is controlled by the combinational logic in Figure 4.10. The multiplican and the three shifted multiplicans are added by the three 12-bit adder banks each composed of three 74LS283 4-bit adders (f53, f45, f37, f54, f46, f38, f55, f47, and f39). The eight most significant bits and the carry are used in Figure 4.12.

The carry bit, if high, indicates overflow and causes chips f40 and f48 to select all ones. Chip f28 uses L, which comes from edge connectors L, M, and N, to set one of the outputs low. This and the AND gates of f29 cause all but the L most significant bits (MSBs) to be set to one by the NAND gates of f31 and f32. The L MSBs are inverted by the NAND gates. Chip f24 checks for an all ones condition. If it exists, chip f21 prevents the EXCLUSIVE OR of chips f22 and f23 from reinverting the L^{th} MSB. This prevents an output of all zeros. Chips f12 and f13 set the output to 10000000 when the channel is inactive. This keeps the prediction at zero waiting for the first sample. Next, f20, f19 and f18 change the $L+1^{th}$ MSB from a zero to a one to take care of truncation. Finally, the group of gates (f17, f18, f25, f26, f27) at the lower left of the schematic convert the three least significant bits of the range number into the proper shifting controls for the multiplier.

The RECONSTRUCT DIFFERENCE-ADAPT RANGE card is shown in Figures 4.13 and 4.14. The input is the 8-bit modified send from the previous card. The three pairs of multiplexers (g9 & g1; g10 & g2; g18 & g17) perform the required shifting for the multiplication. The three sets of adders (g19, g11 & g3; g20, g12 & g4; g21, g13 & g5) add these sums to get the product. This product is shifted right either zero or four places by the four multiplexers (g30, g22, g14 & g6). Figure 4.14 shows the next two multiplexer banks where the

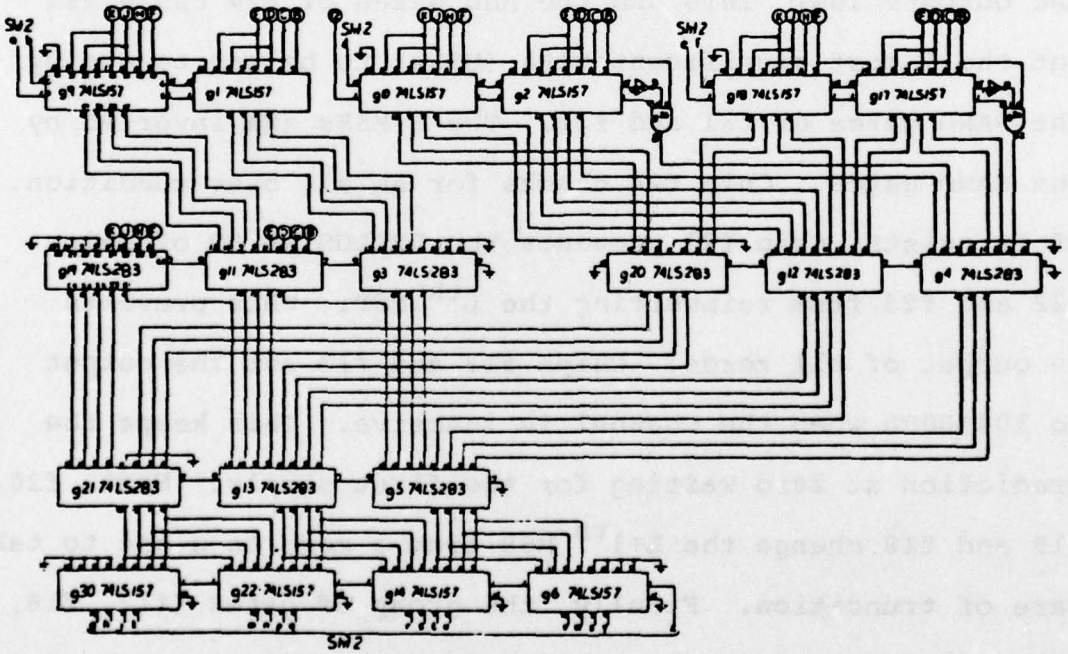


Figure 4.13 Reconstruct difference - adapt range

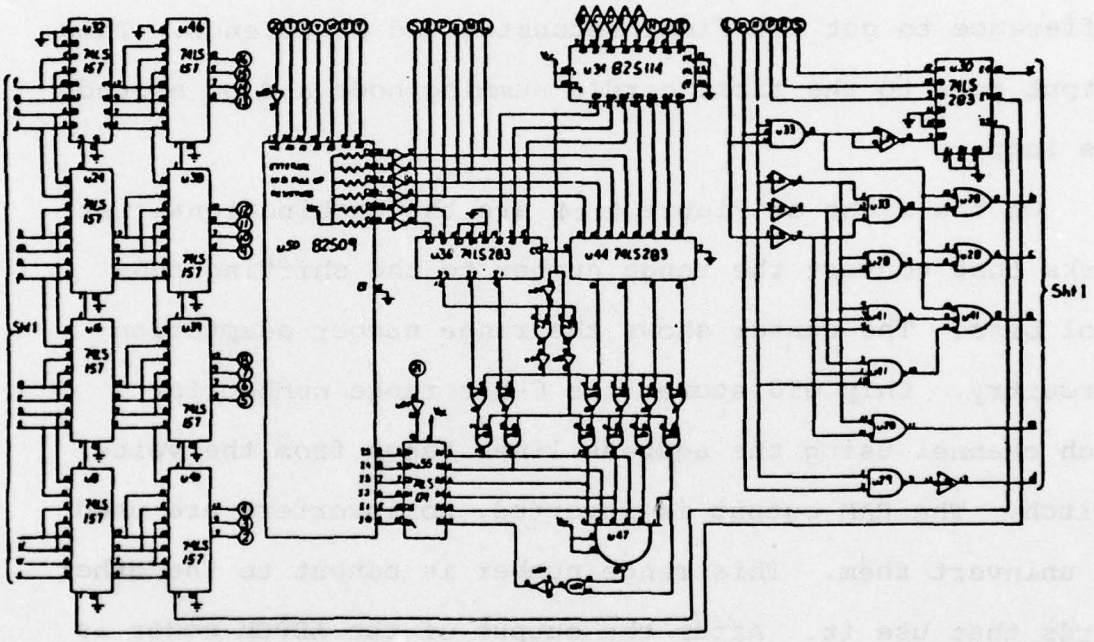


Figure 4.14 Reconstruct difference - adapt range

product is shifted two or zero (g31, g23, g15 & g7) and one or zero (g32, g24, g16 & g8). The result is a 15-bit reconstructed difference mapped from zero to the range size. A ribbon jumper cable carries this output to the "CLAMP DIFFERENCE CARD" of Figure 4.10. Here the upper range limit is subtracted by inverting it and adding it to the difference to get the final reconstructed difference. This output goes to the first card's summing node and on around the loop.

On the right of Figure 4.14 are the combinational networks that convert the range number to the shifting control bits. The center shows the range number adaptation circuitry. Chip g50 stores the 6-bit range number for each channel using the address lines taken from the voice switch. The RAM output is inverted, so inverters are used to uninvert them. This range number is output to the other cards that use it. After the output of the ADPCM coder is valid, it is used to address chip g34, a PROM, whose output is the range number adjustment which varies from -1 to 14. Chips g36 and g44 add the current range number and the adjustment to get a new range number. The Σ3 bit on g36 is the overflow/underflow indication. If a negative adjustment causes underflow, the range number is set to all zeros. If a positive number causes an overflow, the range number is set to all ones (63). Chip g46 detects all ones (63, an invalid range number) and changes the least significant bit

to zero (62). Chip g38 latches the new range number and the RAM stores it for the next sample. The two ADPCM decode cards are virtually identical in function and design to the RECONSTRUCT DIFFERENCE ADAPT RANGE card and the second part of the CLAMP DIFFERENCE TO RANGE card. They differ only in pinouts and the inclusion of the L+1 bit inversion of the second transmit card. They are shown in Figures 4.15, 4.16 and 4.17.

4.3 CAI Error Correcting Code

The use of TASI requires the transmission of Channel Assignment Information (CAI) to the decoder. There it is used to determine the frame format and assign received channels to the proper outputs. Errors in the CAI at the decoder will cause misinterpretation of the frame and faulty channel assignment. This will completely disrupt the active channels until the errors are corrected by the retransmission of the CAI. Since the CAI is transmitted every 3 ms, each error will cause a 3 ms burst of noise in the active channels. The objective is to be sure that CAI errors do not limit the performance of the system.

The projected effect of various transmission error rates on the S/N ratio of the voice channels is shown in Figure 4.18 from [14]. To prevent the CAI errors from limiting performance, errors must be very infrequent at transmission error rates below 10^{-3}. At error rates greater than 10^{-3}, the voice channels will be so noisy that

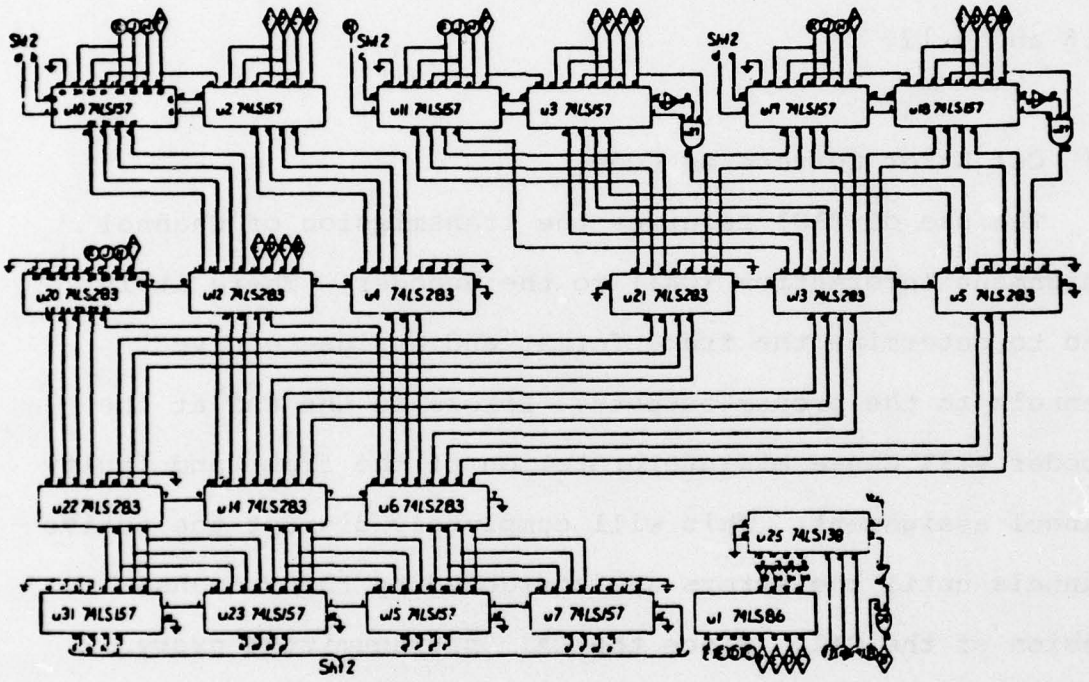


Figure 4.15 Receive - reconstruct difference

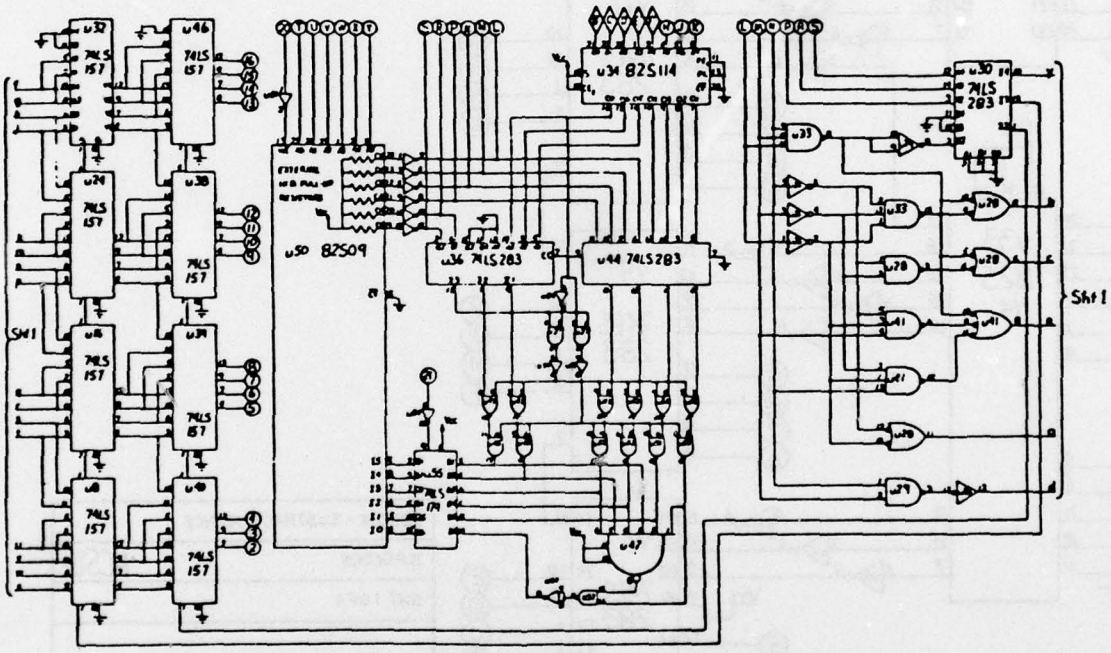


Figure 4.16 Receive - reconstruct difference

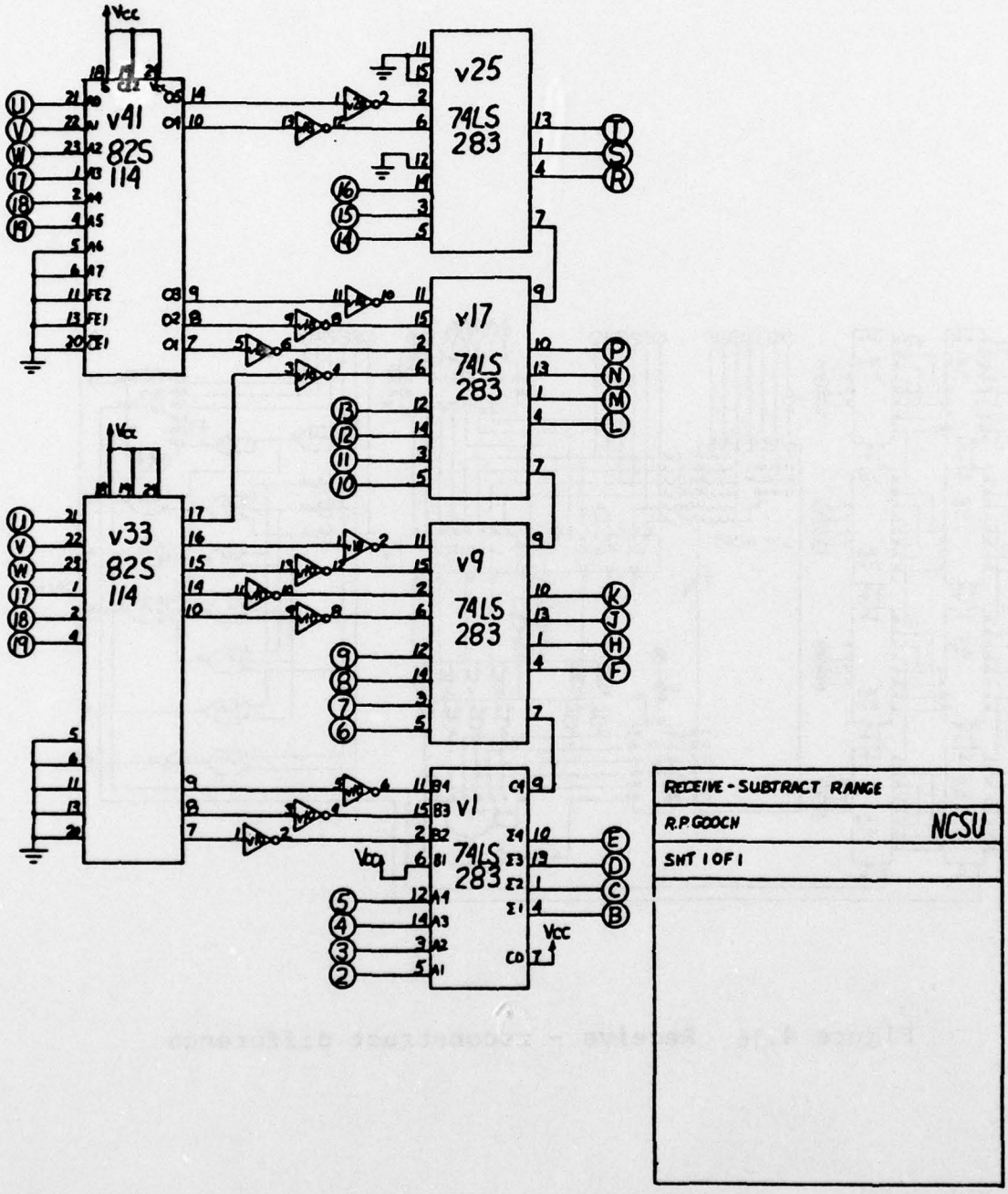


Figure 4.17 Receive - subtract range

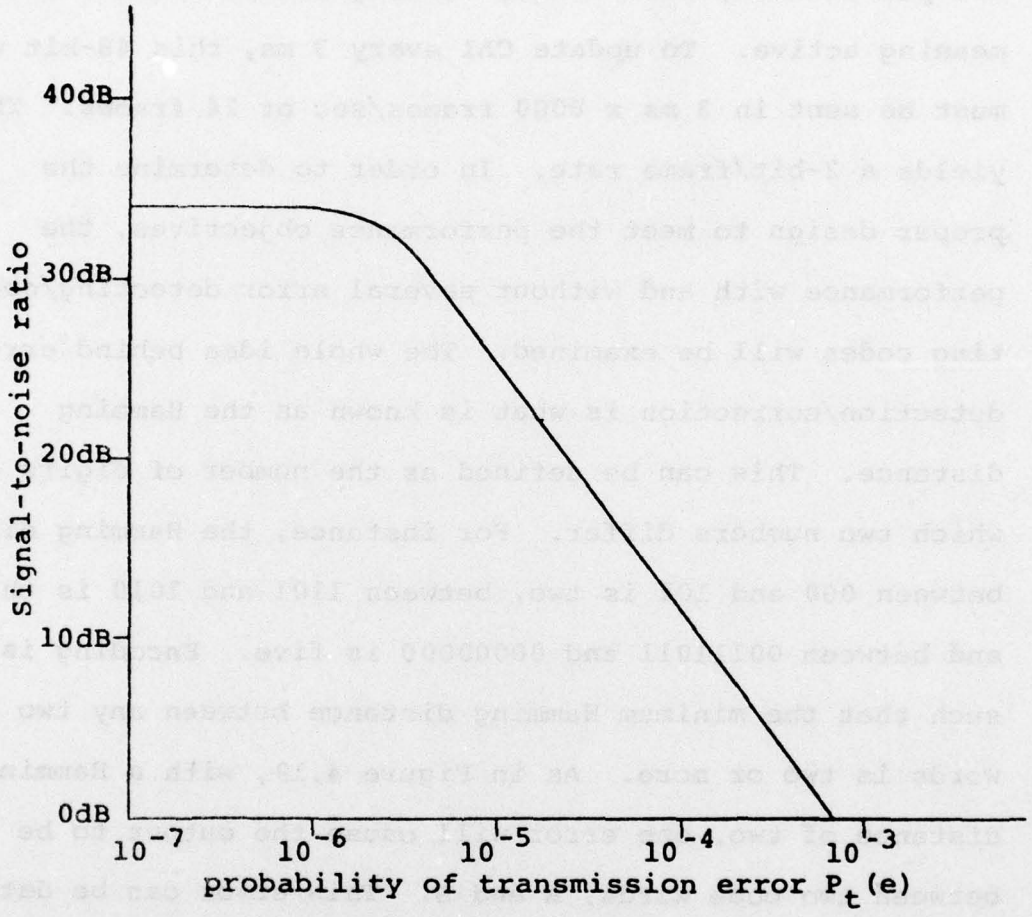


Figure 4.18 Plot of projected signal-to-noise ratio for DDR system versus the transmission error rate

occasional 3 ms noise bursts should remain unnoticed. In the simplest case, the CAI can be sent as a 48-bit word, one bit per channel, with a zero meaning inactive and a one meaning active. To update CAI every 3 ms, this 48-bit word must be sent in 3 ms x 8000 frames/sec or 24 frames. This yields a 2-bit/frame rate. In order to determine the proper design to meet the performance objectives, the performance with and without several error detecting/correcting codes will be examined. The whole idea behind error detection/correction is what is known as the Hamming distance. This can be defined as the number of digits in which two numbers differ. For instance, the Hamming distance between 000 and 101 is two, between 1101 and 1010 is three, and between 00111011 and 00000000 is five. Encoding is done such that the minimum Hamming distance between any two code words is two or more. As in Figure 4.19, with a Hamming distance of two, one error will cause the output to be between two code words, A and B. This error can be detected, but how to correct it cannot be determined. Two errors added to a code word will give a different code word. Thus, two errors cannot be detected with a Hamming distance of two. A Hamming distance of four will allow correction of a single error and detection of two errors. Clearly, the ability of a coding system to detect/correct errors is a function of the minimum Hamming distance. While the usefulness of error correction is evident, the usefulness of

Hamming Distance	# Errors Detected	# Errors Corrected
1	0	0
2	1	0
3	1	1
4	2	1
5	2	2
6	3	2
⋮	⋮	⋮
n	$\text{INT}(\frac{n}{2})$	$\text{INT}(\frac{n-1}{2})$

H.D.=2

0000	VALID CODEWORD A
0010	ERROR
0011	VALID CODEWORD B

H.D.=4

000000	VALID CODEWORD A
010000	CORRECT TO A
010100	2 ERRORS DETECTED
010110	CORRECT TO B
010111	VALID CODEWORD B

Figure 4.19 Code performance as a function of Hamming distance

error detection must be shown. Table 4.2 [15] lists the probabilities that, during a conversation between speaker A and B, there will be a transition from one state to another in a 5 msec period. The four possible states are: 1-speaker A only talking (A), 2-speaker B only talking (B), 3-both A and B talking (T), and 4-neither A nor B talking (N). The entries in Table 4.2 represent the probability that, given the current state, N, A, B, or T, the next state will be N, A, B, or T.

Table 4.2 Probabilities of state transitions during a conversation

From	To	Neither(N)	A	B	Both(T)
Neither	(N)	.98940	.00529	.00530	.00001
A		.00387	.99486	.00001	.00126
B		.00367	.00000	.99510	.00123
Both	(T)	.00005	.00885	.01015	.98905

For instance, the probability that A will be talking 5 msec after neither were talking is P(A N)=.00529. Next, we wish to find the probability that a certain state is given, for example, the percentage of time neither speakers are talking. From [15] we have Table 4.3.

Table 4.3 Percentage of time various states occur during a conversation

State	Percent of time
Talking (per person)	39.50%
P(T) Both talking	4.49%
p(N) Neither talking	25.01%

Clearly, since 2 x 39.5 + 4.49 + 25.01 = 108.5, these are not mutually exclusive events. Talking (per person) includes both persons talking. To find the mutually exclusive probabilities, one must subtract P(N) and P(T) from 1.0, half the result, and assign that as P(A) and P(B):

$$P(A) = .3525$$
$$P(B) = .3525$$
$$P(N) = .2501$$
$$P(T) = .0449$$

$$P(A)+P(B)+P(N)+P(T) = 1.00$$

One can now convert Table 4.2 from conditional probabilities to absolute probabilities by multiplying each row by the probability of starting in that state. Doing this, we get Table 4.4. Next, we shall star (*) those probabilities that represent a change from A to not A or vice-versa.

Table 4.4 Absolute probabilities of state transitions during a conversation

From	To	Neither	A	B	Both
Neither		.24744894	.001323029*	.001325530	.000002501*
A		.001364175*	.35068815	.000003525*	.00044415
B		.001293675	0 *	.35077275	.000433575*
Both		.000002245*	.000397365	.000455735*	.044044655

Summing the starred probabilities, we have

$$P_{CH_5} = P(\text{change in A in 5 ms}) = .00358$$

$$P_{NC_5} = P(\text{no change in A in 5 ms}) = .99642$$

The probability that channel A will change in 5 msec is .00358. The probability that it will change in 3 msec is approximately 3/5 P_{CH_5} = 3/5 x .00358,

$$P_{CH_3} = .00215,$$

$$P_{NC_3} = .99785.$$

Finally the probability that four independent channels will all remain the same is

$$P_{4NC_3} = (P_{NC_3})^4 = .99143$$

$$P_{4CH_3} = 1-P_{4NC_3} = 0.00857 .$$

Thus, 99 percent of the time, if the CAI is received with detected errors, the previous channel assignments will be correct. Error detection is therefore 99 percent as effective as error correction.

The difficult aspect of error correcting codes is the development of codeword sets with the appropriate Hamming spacing. However, the evaluation of the performance of various codes can be done prior to their development. Coding systems are described as a (n,k) code where n is the number of bits per codeword. Four codes known to exist [16] will be evaluated; a (7,4) correct one code, an (8,4) correct one, detect two code, a (15,7) correct two code, and a (16,7) correct two, detect three code. To evaluate the various coding methods one can compute the probability that one or more codewords in a 24 word group will be wrongly interpreted by the decoder versus the probability of an error in the transmission path ($P_t(e)$).

For reference, the performance of a system without error correcting/detecting will be determined. The probability that any particular CAI bit is in error is simply the probability of a transmission error, $P_t(e)$. The probability that none of the 48 CAI bits are in error is

$$P_{CAI}(\emptyset e) = (1-P_t(e))^{48} .$$

The probability that one or more CAI bits are incorrect is

$$P_{CAI}(e) = 1-P_{CAI}(\emptyset e) = 1-(1-P_t(e))^{48}.$$

The CAI is transmitted every 3 msec. The average length of time between incorrect CAI is

$$T_{CAIe} \approx \frac{3ms}{P_{CAI}(e)} .$$

The values of the probability of CAI errors ($P_{CAI}(e)$) and the mean time between CAI errors (T_{CAIe}) for transmission error rates ($P_t(e)$) between 10^{-1} and 10^{-6} are given in Table 4.5.

Table 4.5 Performance with no encoding

$P_t(e)$	$P_{CAI}(e)$	T_{CAIe}	3ms bursts/min
10^{-1}	.993637	3.02 msec	19872
10^{-2}	.382710	7.84 msec	7654
10^{-3}	.046889	.064 sec	938
10^{-4}	.004789	.626 sec	96
10^{-5}	.000480	6.251 sec	10
10^{-6}	.000048	62.502 sec	1

This system would be marginal at even the lowest error rates.

The (7,4) single error correcting code sends four information bits and three check bits per 7-bit codeword. Again the probability that any particular bit in the codeword is

received in error is the probability of a transmission error $P_t(e)$. The probability of exactly K errors in an N-bit codeword is

$$P_w(Ke) = \frac{N!}{K!(N-K)!} P_t(e)^K (1-P_t(e))^{N-K}.$$

This code will give the correct CAI at the decoder if one or no transmission errors occurred in each of the 12 codewords. The probability of one or no transmission errors in a codeword is

$$\begin{aligned} P_w(\text{correctable}) &= P_w(\emptyset e) + P_w(1e) \\ &= (1-P_t(e))^7 + 7P_t(e)(1-P_t(e))^6. \end{aligned}$$

The probability that the CAI is received correctly (P_{CAIc}) is equal to the probability that all 12 codewords are correctable.

$$P_{CAIc} = (P_w(\text{correctable}))^{12}.$$

It follows that

$$P_{CAIe} = 1-P_{CAIc} = 1-(P_w(\text{correctable}))^{12},$$

$$P_{CAIe} = 1-[(1-P_t(e))^7 + 7P_t(e)(1-P_t(e))^6]^{12}.$$

The mean time between CAI errors (T_{CAIe}) is again

$$T_{CAIe} = \frac{3ms}{P_{CAIe}}.$$

Table 4.6 lists the values of P_{CAIe}, T_{CAIe}, and an interpretation of T_{CAIe} versus the transmission error rate $P_t(e)$.

The (8,4) single error correcting, double error detecting code sends four information bits and four check bits per 8-bit codeword. Again, 12 codewords are required to

Table 4.6 Performance of a (7,4) code

$P_t(e)$	$P_{w(e)}$	P_{CAIe}	T_{CAIe}	
10^{-1}	1.49694×10^{-1}	8.57143×10^{-1}	3.500×10^{-3} sec	3.5 msec
10^{-2}	2.03104×10^{-3}	2.41021×10^{-2}	1.245×10^{-1} sec	124 msec
10^{-3}	2.09301×10^{-5}	2.51132×10^{-4}	1.195×10^{1} sec	12 sec
10^{-4}	2.09930×10^{-7}	2.51916×10^{-6}	1.191×10^{3} sec	20 min
10^{-5}	2.09993×10^{-9}	2.51992×10^{-8}	1.191×10^{5} sec	33 hrs
10^{-6}	2.09993×10^{-11}	2.51992×10^{-8}	1.191×10^{7} sec	138 days

send all 48 bits of CAI. The CAI will be received correctly if each codeword either has zero or one error or has two errors and no change in state of the four channels represented. The probability that a codeword is correctly interpreted is

$$P_w(\text{correctable}) = P_w(\emptyset e)+P(1e)+P(2e)\cdot P_{_4NC_3},$$

where $P_{_4NC_3}=0.99143$.

$$P_w(\text{correctable})=(1-P_t(e))^8+8P_t(e)(1-P_t(e))^7$$
$$+28P_t(e)^2(1-P_t(e))^6\cdot.99143\ .$$

Again, the probability that the CAI is received correctly, P_{CAIc}, is equal to the probability that all 12 codewords are correctable.

$$P_{CAIc}=(P_w(\text{correctable}))^{12}$$
$$P_{CAIe}=1-(P_w(\text{correctable}))^{12}$$
$$P_{CAIe}=1-[(1-P_t(e))^8+8P_t(e)(1-P_t(e))^7$$
$$+28P_t(e)^2(1-P_t(e))^6\cdot.99143]^{12}.$$

Table 4.7 lists these values versus $P_t(e)$.

Table 4.7 Performance of an (8,4) code

$P_c(e)$	$P_w(e)$	P_{CAIe}	T_{CAIe}	
10^{-1}	3.93670×10^{-2}	3.82425×10^{-1}	7.854×10^{-3} sec	7 msec
10^{-2}	7.65251×10^{-5}	9.17915×10^{-4}	3.268×10^{0} sec	3 sec
10^{-3}	2.94314×10^{-7}	3.53176×10^{-6}	8.494×10^{2} sec	14 min
10^{-4}	2.45414×10^{-9}	2.94497×10^{-8}	1.019×10^{5} sec	28 hrs
10^{-5}	2.40506×10^{-11}	2.88607×10^{-10}	1.039×10^{7} sec	4 months
10^{-6}	2.40015×10^{-13}	2.88017×10^{-12}	1.042×10^{9} sec	33 years

The (15,7) double error correcting code sends seven information bits and eight check bits per 15-bit codeword. Seven codewords are required to transmit the 48-bit CAI. The CAI will be received correctly if each codeword has zero, one, or two errors. As before,

$$P_w(\text{correctable}) = P_w(0e) + P_w(1e) + P_w(2e),$$

$$P_w(\text{correctable}) = (1-P_t(e))^{15} + 15P_t(e)(1-P_t(e))^{14} + 105P_t(e)^2(1-P_t(e))^{13},$$

$$P_{CAIe} = 1 - [(1-P_t(e))^{15} + 15P_t(e)(1-P_t(e))^{14} + 105P_t(e)^2(1-P_t(e))^{13}]^7 .$$

Table 4.8 shows this code's performance.

Table 4.8 Performance of a (15,7) code

$P_t(e)$	$P_w(e)$	P_{CAIe}	T_{CAIe}	
10^{-1}	1.84061×10^{-1}	7.59229×10^{-1}	3.951×10^{-3}	4 msec
10^{-2}	4.15802×10^{-4}	2.90699×10^{-3}	1.032×10^{0}	1 sec
10^{-3}	4.50923×10^{-7}	3.15646×10^{-6}	9.504×10^{2}	16 min
10^{-4}	4.54591×10^{-10}	3.18213×10^{-9}	9.428×10^{5}	11 days
10^{-5}	4.54959×10^{-13}	3.18471×10^{-12}	9.420×10^{8}	30 years
10^{-6}	4.54996×10^{-16}	3.18497×10^{-15}	9.419×10^{11}	298 centuries

The (16,7) double error correcting, triple error detecting code sends seven information bits per 16-bit codeword. Seven codewords are required for the CAI. As before,

$$P_w(\text{correctable}) = P_w(\emptyset e) + P_w(1e) + P_w(2e) + P_w(3e) \cdot P_{_7NC_3} ,$$

where $P_{_7NC_3} = .98505$;

$$\begin{aligned} P_w(\text{correctable}) = {} & (1-P_t(e))^{16} + 16P_t(e)(1-P_t(e))^{15} \\ & + 120P_t(e)^2(1-P_t(e))^{14} \\ & + 560P_t(e)^3(1-P_t(e))^{13} \cdot .98505 \end{aligned}$$

$$\begin{aligned} P_{CAIe} = 1-[& (1-P_t(e))^{16} + 16P_t(e)(1-P_t(e))^{15} \\ & + 120P_t(e)^2(1-P_t(e))^{14} \\ & + 551.628\ P_t(e)^3(1-P_t(e))^{13}]^7 \end{aligned}$$

Table 4.9 shows thid code's performance.

Table 4.9 Performance of a (16,7) code

$P_t(e)$	$P_w(e)$	P_{CAIe}	T_{CAIe}	
10^{-1}	7.05343×10^{-2}	4.00715×10^{-1}	7.487×10^{-3}	7 msec
10^{-2}	2.37219×10^{-5}	1.66041×10^{-4}	1.807×10^{1}	18 sec
10^{-3}	1.00664×10^{-8}	7.04650×10^{-8}	4.257×10^{4}	12 hours
10^{-4}	8.49795×10^{-2}	5.94856×10^{-11}	5.043×10^{7}	19 months
10^{-5}	8.38911×10^{-15}	5.87238×10^{-14}	5.109×10^{10}	16 centuries
10^{-6}	8.37371×10^{-18}	4.86160×10^{-17}	5.118×10^{13}	1.6 million years

The mean time between errors T_{CAIe} versus the transmission error rate is shown in Figure 4.20 for no encoding and for the four codes. The figure shows that three codes limit the errors to one every 15 minutes with an error rate

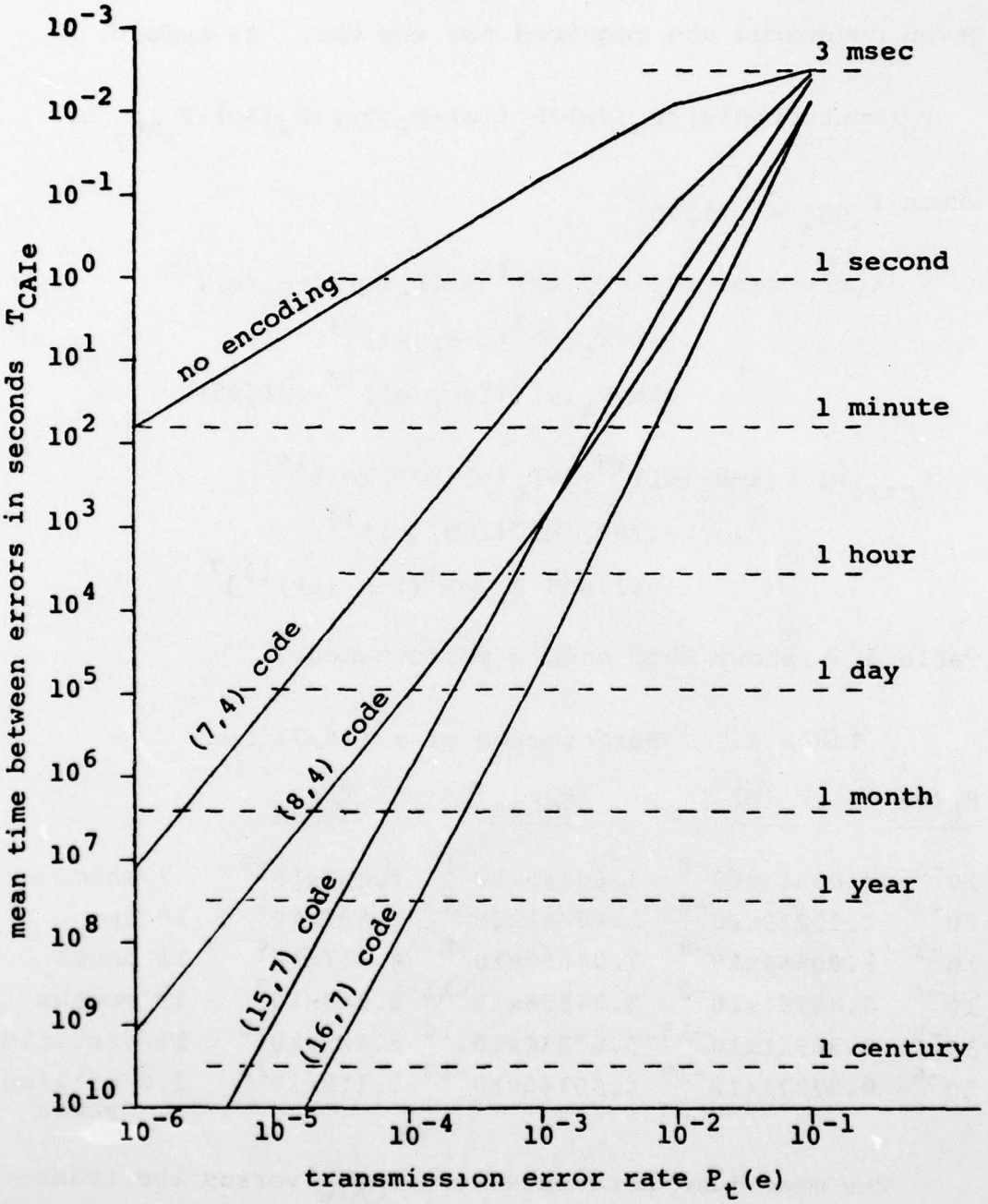
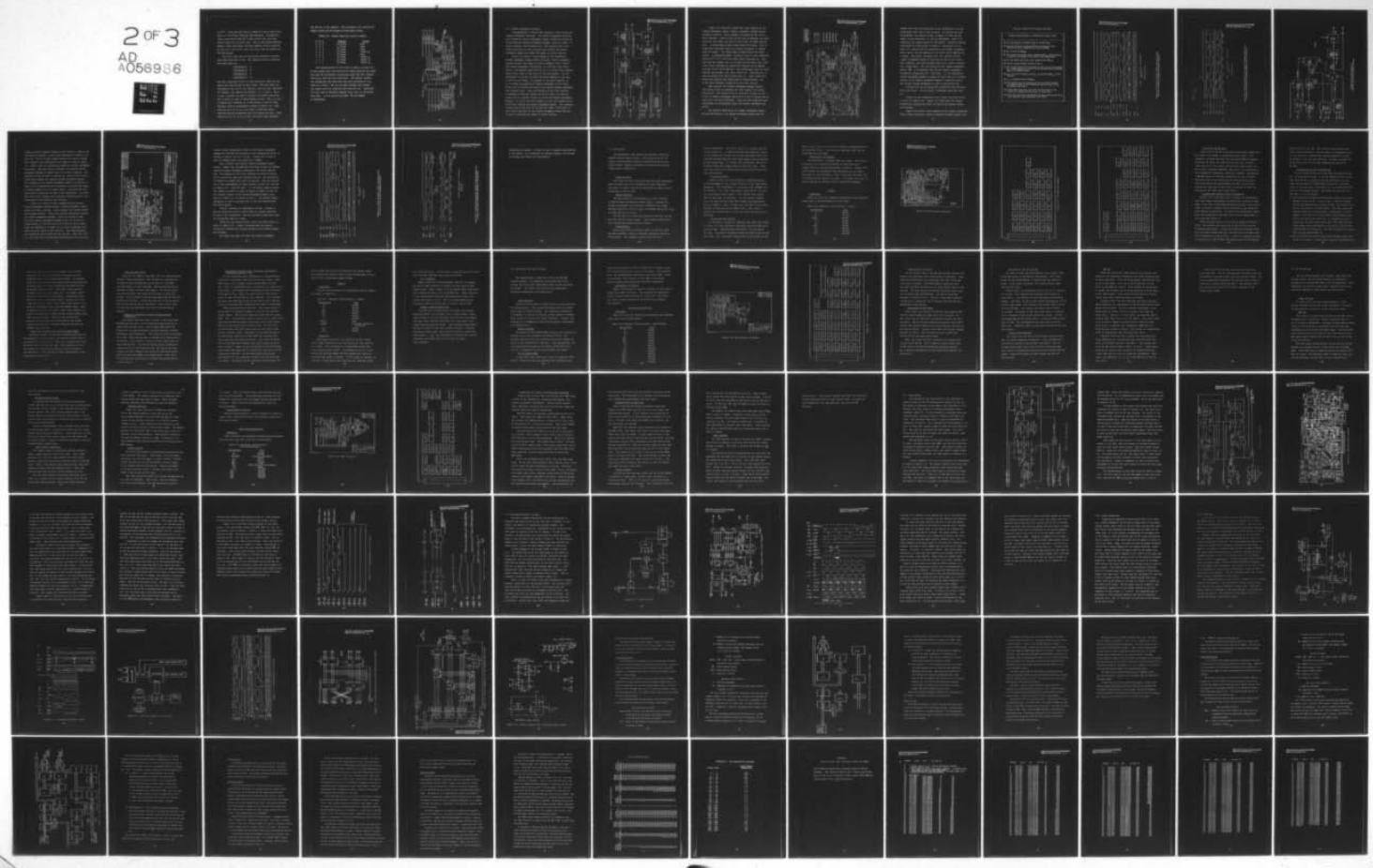


Figure 4.20 Plot of mean time between errors versus transmission error rate, using various encoding methods

AD-A056 986 GEORGIA INST OF TECH ATLANTA F/G 17/2
DIGROUP DATA REDUCTION SYSTEM.(U)
JUN 78 J B O'NEAL F30602-75-C-0118
UNCLASSIFIED RADC-TR-78-139 NL

2 OF 3
AD
A056986



of 10^{-3}. Since the S/N ratio of ADPCM is 0 dB at this error rate, a 3 ms "click" would be imperceptible. Transmission error rates better than 10^{-5} yield errors less than four times a year for the (8,4) code. Due to the longer updating delays, lower efficiency, and more complex circuit required by the (15,7) and (16,7) code, the (8,4) code was chosen for the DDR system.

The (8,4) code used is the correct one/detect two Hamming code described in [12]. The equations used to generate the check bits are:

$$C_1 \oplus I_1 \oplus I_2 \oplus I_4 = 0 \quad A$$

$$C_2 \oplus I_1 \oplus I_3 \oplus I_4 = 0 \quad B$$

$$C_3 \oplus I_2 \oplus I_3 \oplus I_4 = 0 \quad C$$

$$C_4 \oplus I_1 \oplus I_2 \oplus I_3 = 0 \quad D.$$

The four information bits are sent during one frame and the check bits sent during the next frame. When the check and information bits are at the receiver, the same four equations are formed. The results are called A, B, C, and D. The A, B, and C results are used to correct the single errors and the D result is used to detect double errors. The A, B, and C results are connected to a 3-bit binary to one of eight decoders which is programmed as shown in Table 4.10. When only one of A, B, C, or D is a one, a check bit error occurred and the information bits are correct as sent. When three of A, B, C, or D is a one, two errors have occurred

and the CAI is not updated. This procedure will correct all single errors and 99 percent of the double errors.

Table 4.10 Action taken for various results

A	B	C	Meaning	Action
0	0	0	No errors	None
0	0	1	C_3 wrong	None
0	1	0	C_2 wrong	None
0	1	1	I_3 wrong	Invert I_3
1	0	0	C_1 wrong	None
1	0	1	I_2 wrong	Invert I_2
1	1	0	I_1 wrong	Invert I_1
1	1	1	I_4 wrong	Invert I_4

The implementation of this code is shown in Figure 4.21. On the encode side, the exclusive-or gates generate the check bits and the multiplexer alternately sends the four information bits, then the four check bits. The decoder latches the information and check bits and uses exclusive-or's to check for errors. The one-of-eight decoder will correct any single error by inverting the incorrect bit. Exclusive-or's are used to determine whether zero, one, or two errors were made. If two errors were made, the CAI update is suppressed.

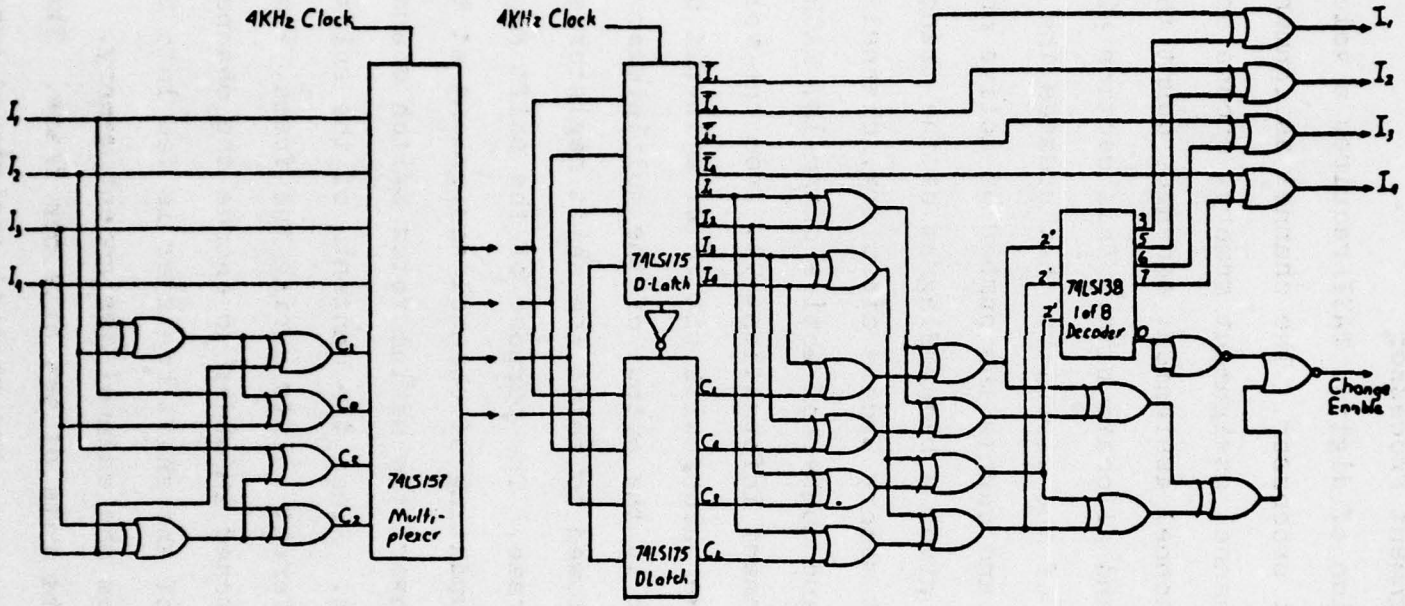


Figure 4.21 CAI error correcting code implementation

4.4 Channel Assignment Processor

Implementation of digital TASI requires a voice switch and channel assignment processor. The channel assignment processor must update the channel assignment memory, encode and decode the error-protected channel assignment message, count the number of active channels, and allocate bits. This section gives a detailed discussion of the circuitry that updates the channel assignment memory and counts the number of active channels.

Figure 4.22 gives a block diagram of the transmitter channel assignment memory update circuitry, channel assignment message encoder, and number of active channels, NACH, counter. New channel assignment information, CA, from the voice switch enters a multiplexer along with bit 16 of a 16-bit shift register whose serial input is the output of the multiplexer. The new information is allowed to enter the shift register at the rate of two bits per frame. The purpose of the shift register is: (1) to delay the updating of channel assignment at the transmitter and (2) convert the serial voice switch channel assignment into parallel form. Since the contents of the shift register is updated at different points during the frame, its parallel output must be latched in order to encode the channel assignment message. Bit 16 of the shift register is fed into another multiplexer that updates the channel assignment memory. This updating is also done at the rate of two bits per frame. The output of this multiplexer goes to the channel assignment memory RAM and is used in counting the number of active channels.

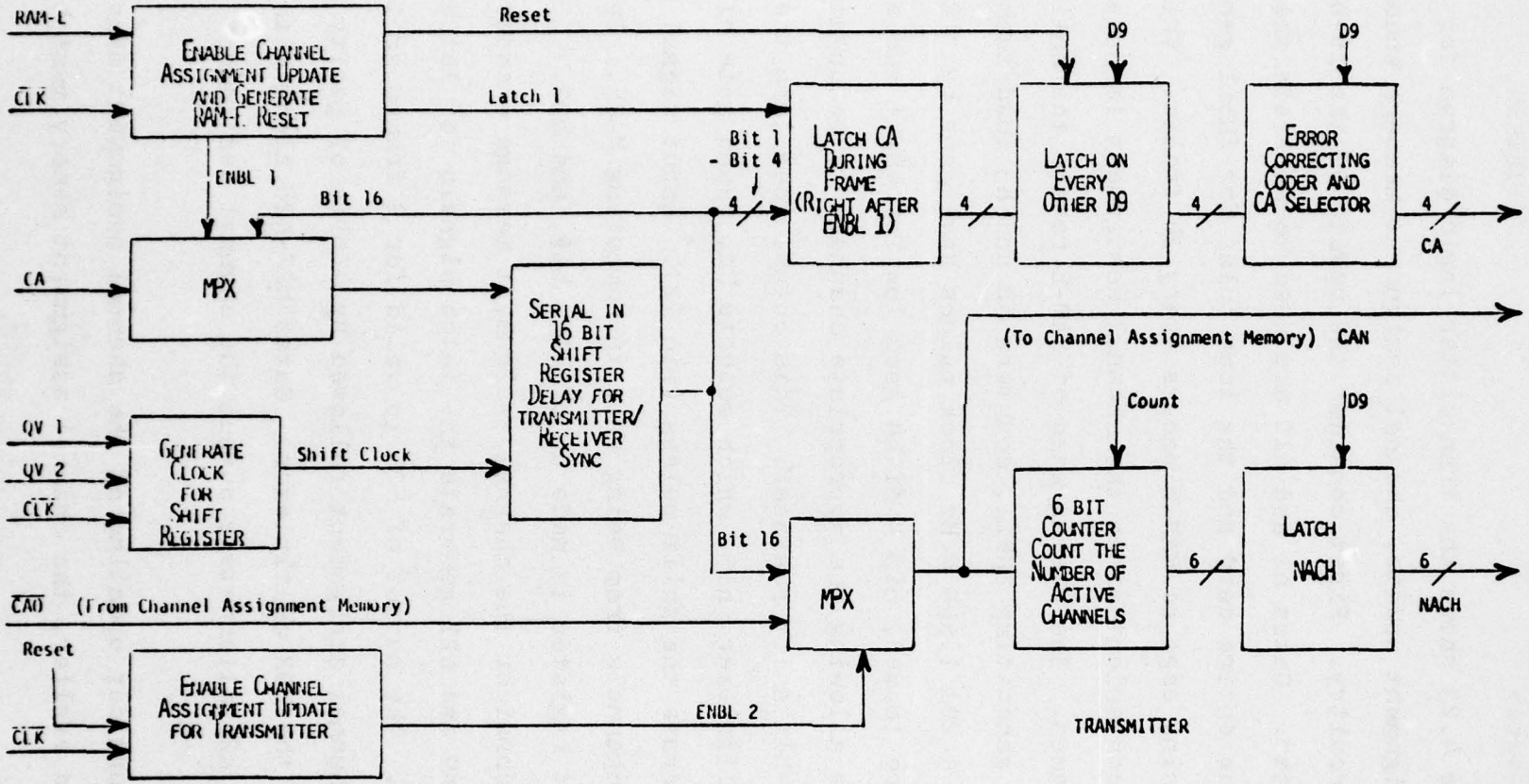


Figure 4.22 Block Diagram of Transmitter Channel Assignment Processor

Figure 4.23 shows the transmitter logic diagram for the channel assignment update, channel assignment message encoder, and NACH circuitry. First consider the update of the 16-bit shift register. Gates h1 and h10 are used to generate the reset pulse for the entire card and the load pulse for ENBL1 generation. A rising edge of RAM-E occurs every 24 frames. This is especially convenient since the channel assignment is updated every 24 frames. The rising edge of RAM-E resets the entire system. In generating ENBL1, counters h2 and h3 load themselves after 193+8 = 201 1.544 M Hz clock pulses have gone by. Each time they are loaded, pin 4 of h4 goes low for eight and a half clock pulses allowing the appropriate channels to be updated into the 16-bit shift register. This output goes into the shift register multiplexer, h13, which selects CA. Nand gate h11 is used to generate the shift pulses (pin 12). Input signal DCS keeps data channels from being used in computing NACH. The 16-bit shift register is made up of h12, h19, and h37.

Next consider the channel assignment message encoder. Flip-flops h9 and h21 generate the latch signals for latches h29 and h20. The output of h20 is valid for 2 frames so that 4 bits of channel assignment followed by 4 bits of parity may be sent out by the DDR multiplexer. Gates h27-h28 along with quad multiplexer h26 alternately output the channel assignment and parity.

Now consider updating of the channel assignment memory. As mentioned earlier, the channel assignment memory must be

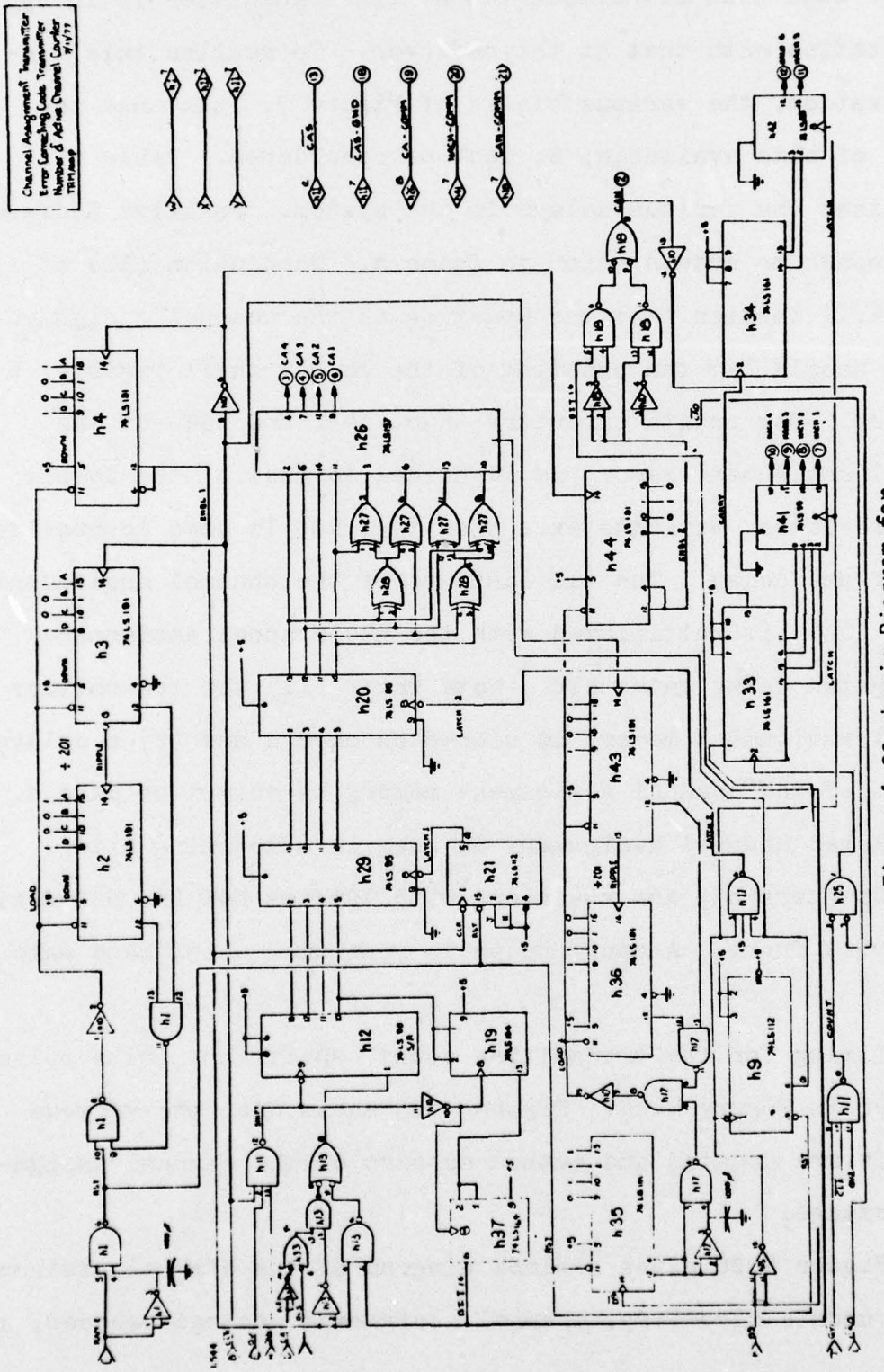


Figure 4.23 Logic Diagram for
Transmitter Channel
Assignment Processor

updated such that bit allocation at the transmitter is in synchronization with that at the receiver. To realize this synchronization, the various blocks of Figure 3.3 that use the number of bits available, B, must be considered. Table 4.11 summarizes the various delays in the system. Positive indexing corresponds to events prior to frame n. Conclusion (12) of Table 4.11 implies that the updating of the channel assignment memory should lag the updating of the 16-bit shift register by 6 frames. The update circuitry--h36, h43, and h44--of the channel assignment memory is identical to that of the 16-bit shift register, with the exception that h35 is used to provide the 6 frame delay. The old contents of the channel assignment memory, $\overline{CAO}$, is multiplexed with the new channel assignment information using gates h18. Note that: (1) the transmitter channel assignment memory is stored on card d and (2) a delayed version of the channel assignment memory is output by card d. The delayed channel assignment is used in allocating bits.

Counters h33 and h34 along with latches h41 and h42 derive NACH every frame. A count pulse is generated using Nand gate h11.

Timing for the transmitter count, shift, and write pulses is given in Figure 4.24. Figure 4.25 shows when the various channels are updated and output as part of the channel assignment message.

Figure 4.26 gives a block diagram of the channel assignment memory update circuitry, channel assignment message decoder, and

Table 4.11 Dynamic Bit Allocation and Delay

Suppose the Multiplexer is shifting out frame n, then:
(1) The multiplexer is forming frame n-1 based on B_n
(2) The bit allocator at the transmitter outputs B_n to the multiplexer while computing L_{n-1} based on $NACH_n$
(3) B_n is based on $NACH_{n+1}$
(4) The channel assignment update computes $NACH_{n-1}$ while updating 2 channels (the update is included in the computation)
(5) (1)-(4) imply that frame n was formatted per $NACH_{n-2}$
(6) The bit stream decoder receives frame n
(7) (6) implies that the bit allocator at the receiver must output B_{n+1} asynchronously to the bit stream decoder while outputting B_{n+2} to the ADPCM decoder
(8) B_{n+1} is based on $NACH_{n+2}$ and B_{n+2} is based on $NACH_{n+3}$ at the receiver
(9) L_n is computed based on $NACH_{n+1}$
(10) (9) implies that the updating of the transmitter channel assignment memory at the transmitter must lead that of the receiver by one frame
(11) After RAM-E goes high, the first possible update at the receiver is channels 39 and 40 due to frame delay
(12) (10) and (11) imply that channels 37 and 38 must be the first to be updated at the transmitter after RAM-E

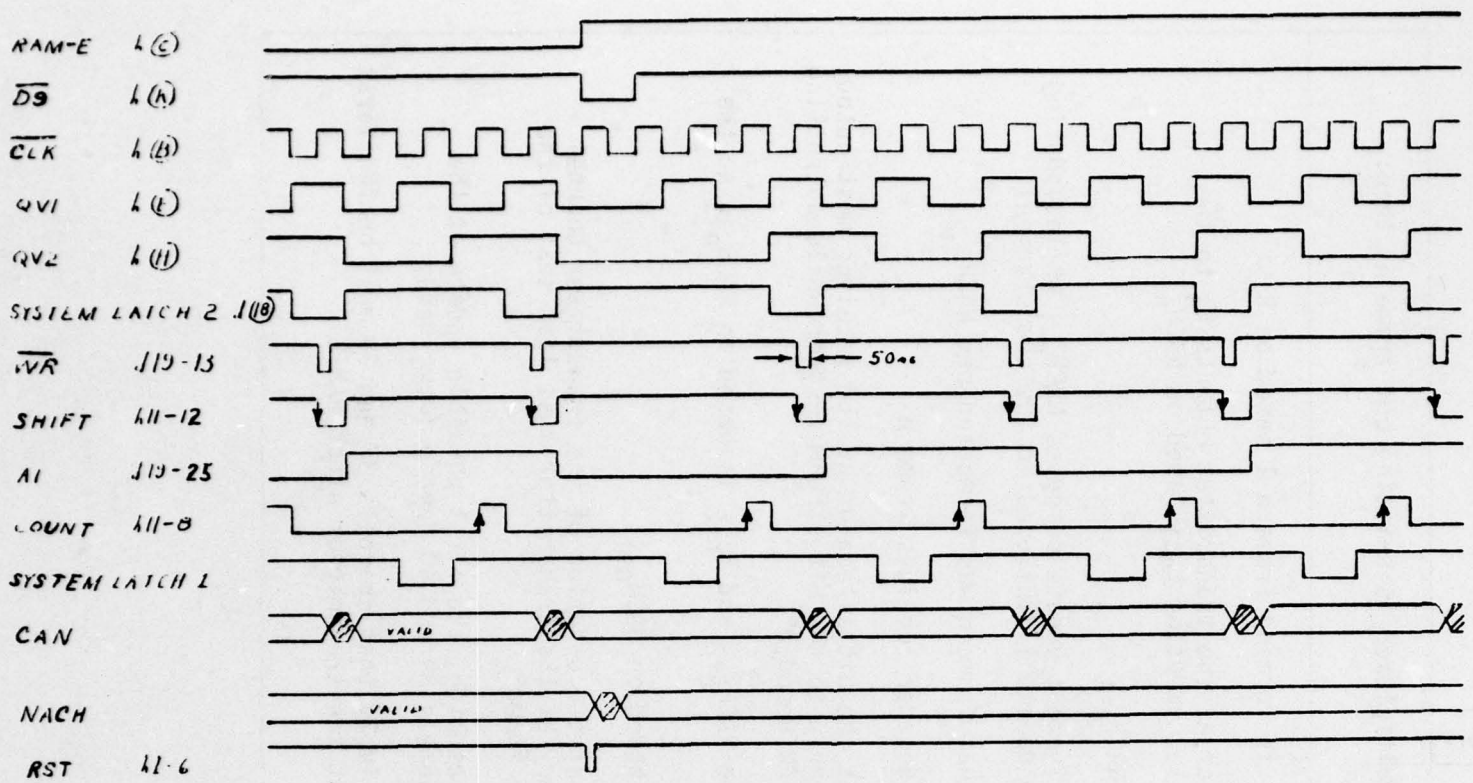
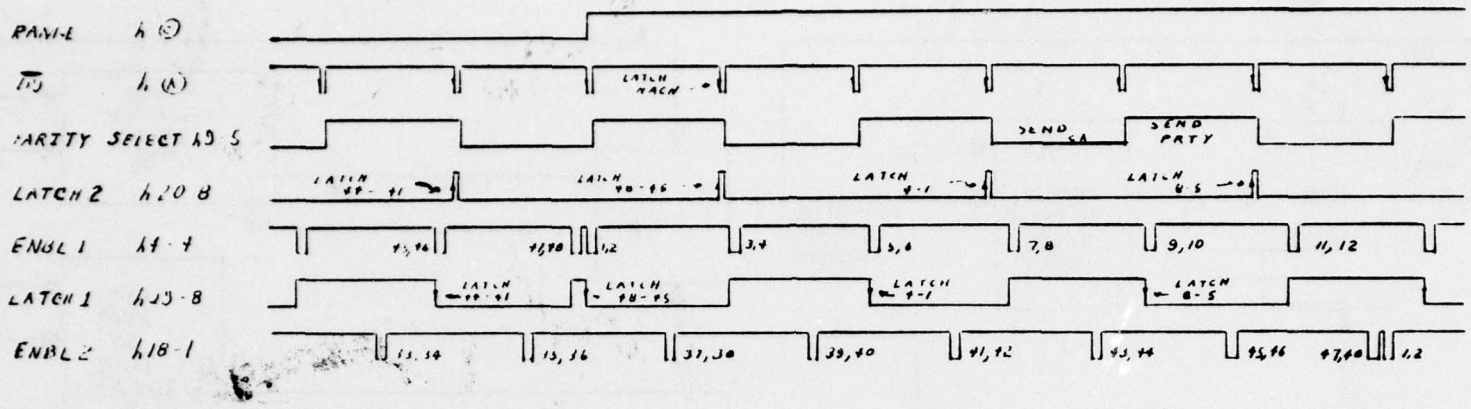


Figure 4.24 Timing for Count, Shift, and Write Pulses for the Transmitter Channel Assignment Processor



TIMING FOR CHANNEL ASSIGNMENT
CARD - 4

TRANSMITTER

Figure 4.25 Timing for Updating the Channel Assignment Memory and Latching the Channel Assignment Message at the Transmitter

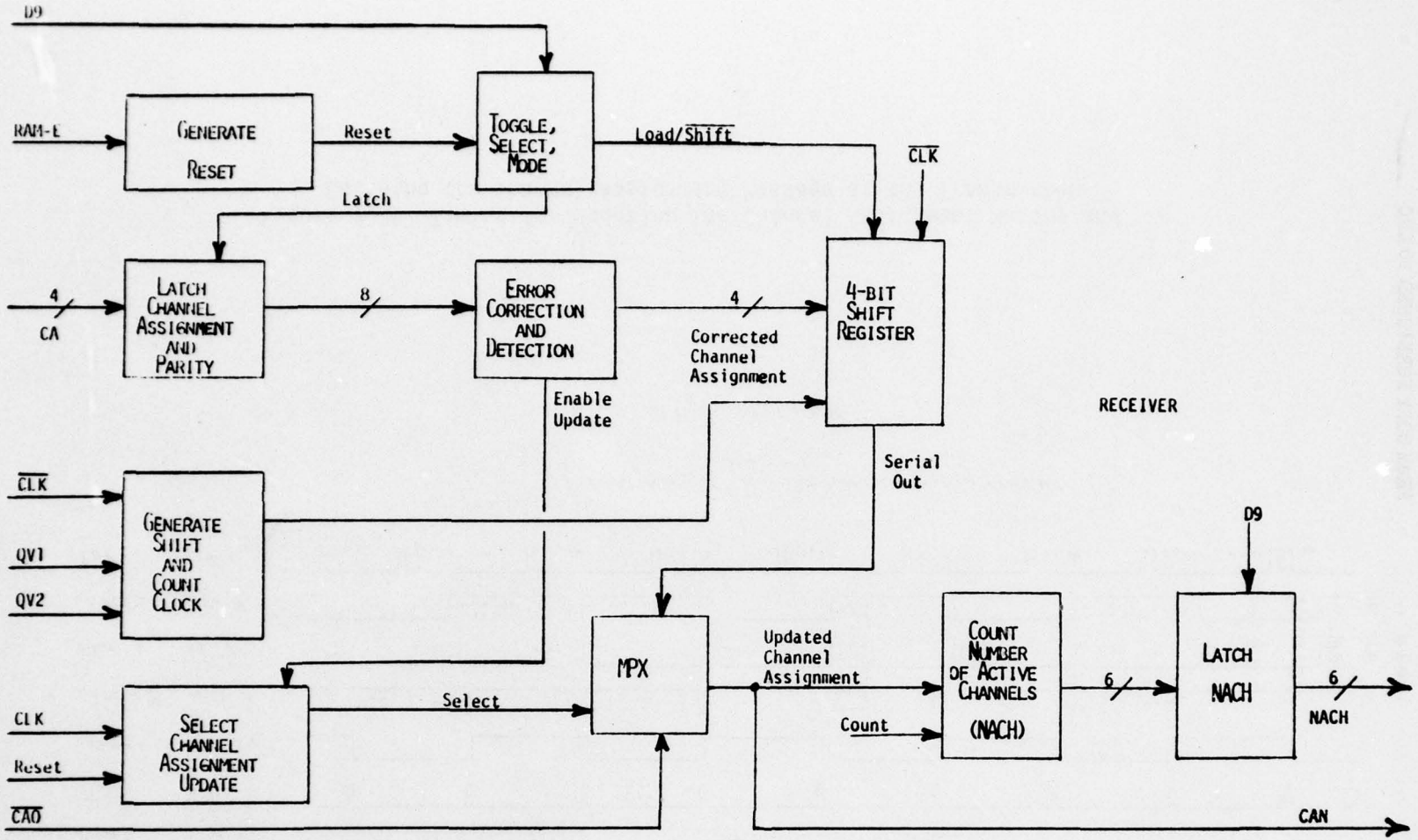


Figure 4.26 Block Diagram of
Receiver Channel Assignment Processor

number of active channels counter at the receiver. Many of the functions are similar, as seen from comparison of Figures 4.22 and 4.26. The bit stream decoder receives the channel assignment message at the beginning of the frame as shown in Figure 2.6. These 4 bits are alternately latched as channel assignment and parity. Once the parity is latched, the decoded channel assignment message is loaded into a 4-bit shift register. The contents of this register is shifted out serially into a multiplexer that updates the receiver channel assignment memory. This updating of the channel assignment memory is identical to that of the transmitter with the exception of having the number of frames delayed set at 5 rather than 6. The circuitry that counts NACH is identical to that of the transmitter. Note that the 16-bit shift register and associated circuitry used at the transmitter is not needed at the receiver.

Figure 4.27 gives the logic diagram for the receiver channel assignment message decoder, channel assignment memory update, and NACH circuitry. First consider the channel assignment message decoder. Latches s41 and s49 alternately capture the channel assignment message. Parity is latched in s49. Since the channel assignment message as put out by the bit stream decoder is not valid until 12 1.544 M Hz clock pulses after the beginning of a frame, s12 is used to generate a delayed clock for input to flip-flop s27. This flop generates the appropriate latch pulses for s41 and s49. Gates s42, s43, s50, s51, s33, and s35 along with demultiplexer s34 detect and

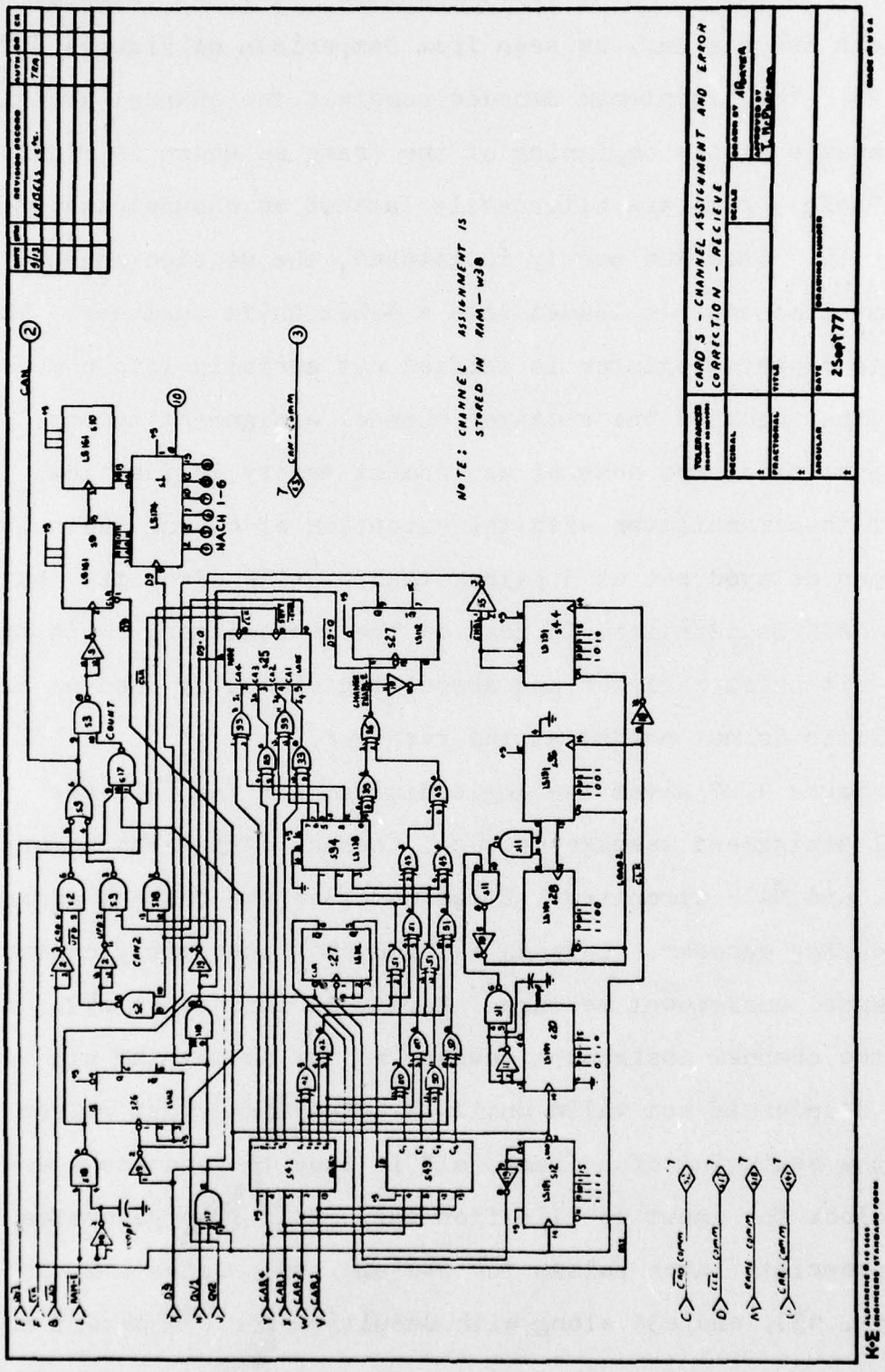


Figure 4.27 Logic Diagram for
Receiver Channel Assignment Processor

correct single transmission errors in the channel assignment message and indicate the detection of two transmission errors by placing a logical 0 on pin 1 of s35. D-latch s27 is used to hold the Change Enable line valid for 2 frames.

Next, consider the receiver channel assignment memory update. Toggle flop s26 generates the proper pulses for loading corrected channel assignment information into shift register s25. The contents of this shift register is serially shifted into a multiplexer made up of gates s2 and s3. The multiplexer is enabled by counters s20, s28, s36 and s41. Note the similarity of the configuration of these counters to h36, h43, and h44 of Figure 4.23. Also note that: (1) the actual channel assignment memory is stored on card w as shown in Figure 4.4 and (2) two delayed versions of the channel assignment memory, also shown in Figure 4.4, are output by card w. The delayed channel assignment is used to allocate bits to the bit stream decoder and ADPCM decoder.

Finally, consider the computation of NACH. Counter s9 and s10 along with latch s1 derive NACH in a manner identical to that of the transmitter. The receiver NACH output here leads the transmitter NACH by 1 frame.

Timing for the receiver count, shift, and write pulses is given in Figure 4.28. Figure 4.29 shows when the various channels are updated and received as part of the channel assignment message.

The timing and order of events for channel assignment

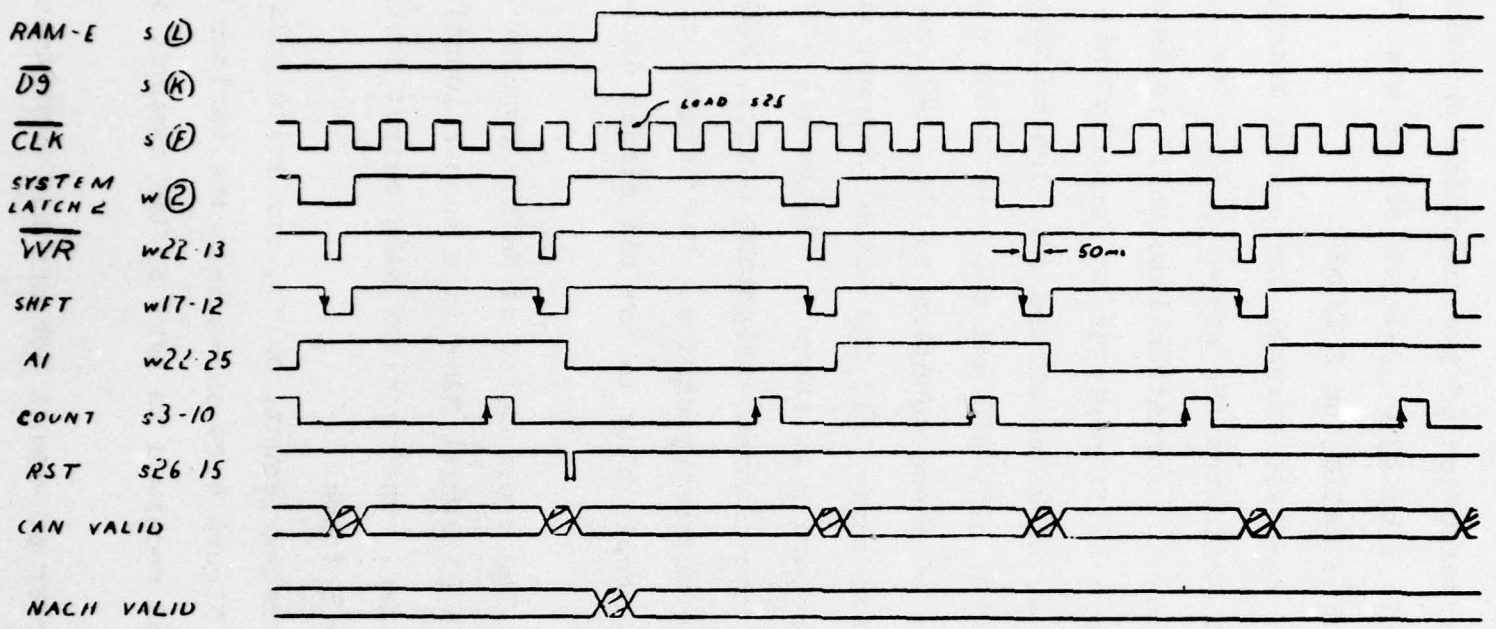
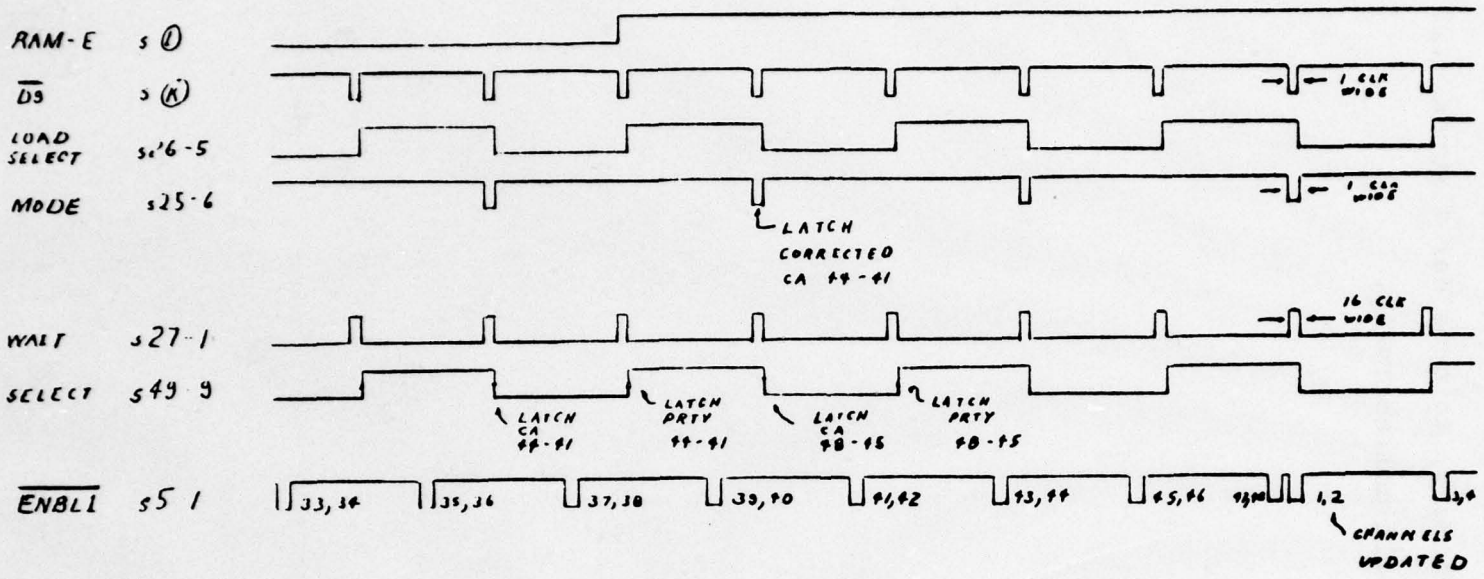


Figure 4.28 Timing for Count, Shift, and Write Pulses for the Receiver Channel Assignment Processor



TIMING FOR CHANNEL ASSIGNMENT
CARD - 6
RECEIVER

Figure 4.29 Timing for Updating the Channel Assignment Memory and Latching the Channel Assignment Message at the Receiver

processing is complex. In order to get a complete understanding of the system, it is necessary to consider figures 4.22 through 4.29 along with Table 4.11 collectively.

4.5 Multiplexer

The multiplexer runs totally on external timing and handles several types of data. These data are L-bit VF words, 8-bit DD words, signaling information (8 bits per output frame), channel assignment information (4 bits per frame), and a framing bit.

Input sections

All input sections associated with the above mentioned types of data consist of 74LS166 8-bit shift registers. This data is loaded in parallel and passed to other circuit components in serial form.

Output sections

The final gate of the multiplexer, a DF-F (74LS74), is fed by three multiplexer output stages. Framing bit, channel assignment, and signaling information are right shifted from SR-2, SR-3 of stage 1 (74LS166) during the first 13 clock pulses of a frame.

VF, DD, and 1-fill data are shifted to the DF-F by two parallel 180-bit output stages. These output stages alternate loading and outputting every frame.

Timing delays

Due to the several different types of circuits used for data transfer, several situations required a delay of clock pulses. For example, suppose there are two

circuit components. The first, say F-F, is to pass one bit to the second F-F. If these two flops are identical, there is no problem; but if the circuits are different, the first flop may not hold the bit valid long enough to be recognized by the second flop. To correct such situations, one would slightly delay the trigger pulse to the first flop. Therefore, the second flop would latch the data before the first flop could change.

Position markers

Several facets of the multiplexer's operation depend on the knowledge of the current position in the frame being outputted. Such information is provided by the MIN/MAX pins of 74LS191's. As a new frame begins, each 74LS191 is reset to a particular binary number. Each 74LS191 then counts down as the frame is outputted. As each counter reaches zero, its MIN line goes high and triggers the appropriate circuit elements. Note, in most cases the MIN line of each 74LS191 is also tied to its ENABLE line. Thus, once a counter has reached zero, it will not count for the rest of the frame.

T/S of 2518 HEX 32 S/R

As the DDR is running at 3.088 MHz (two times the normal T1 rate), all data-associated components must be rated at at least 3.5 MHz. Unfortunately, the 2518's are only rated at 2 MHz. Therefore, instead of using one 2518 in the serial data flow, use a parallel combination of two 2518's in the

data stream. Bits are alternately loaded and outputted from the parallel 2518's. This gives an apparent clock rate of 1.544 MHz for each 2518.

Explanation of diagrams

The multiplexer is broken into two stages. For ease of reference, all associated drawings are presented here. Figure 4.30 is an overall multiplexer schematic. Figures 4.31 and 4.32 give the location of the components on the stage 1 and stage 2 circuit boards. Stage 1 comprises the upper half of Figure 4.30 and stage 2 the lower. The reader can refer to these drawings as necessary while reading this chapter.

Stage 1

Components

Table 4.12 lists the component designations and component types used in the multiplexer's first stage.

Table 4.12 Component identification - stage 1

Designation	Type
A__	74LS08
C__	74LS191
COMP__	74LS85
D__	74LS74
I__	74LS04
NA__	74LS00
O__	74LS32
S__	74LS123
SR__	74LS166

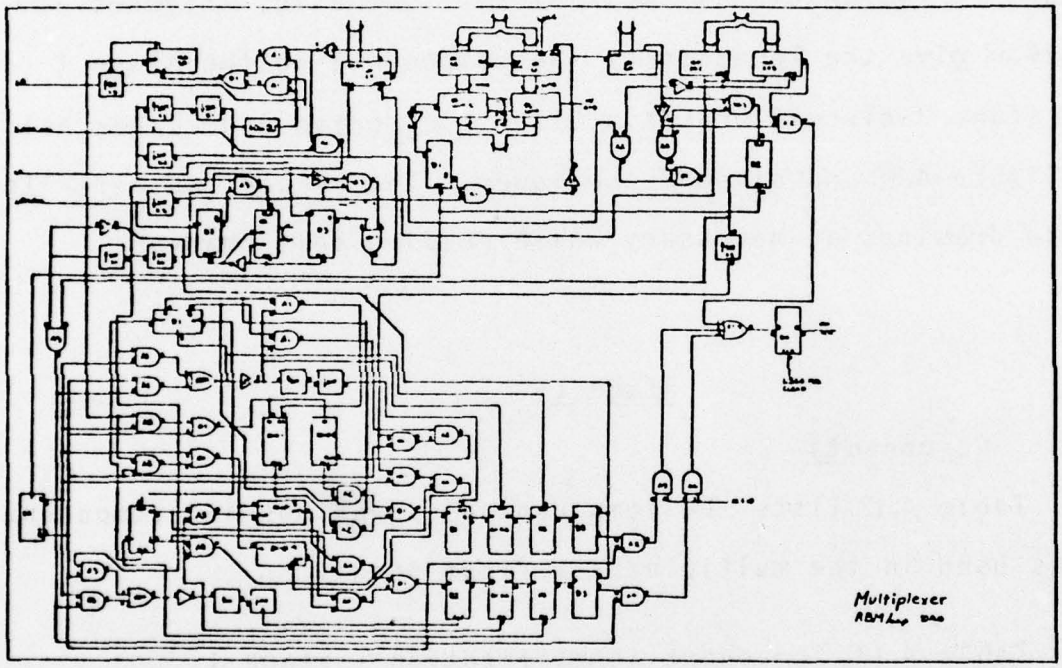


Figure 4.30 Multiplexer schematic

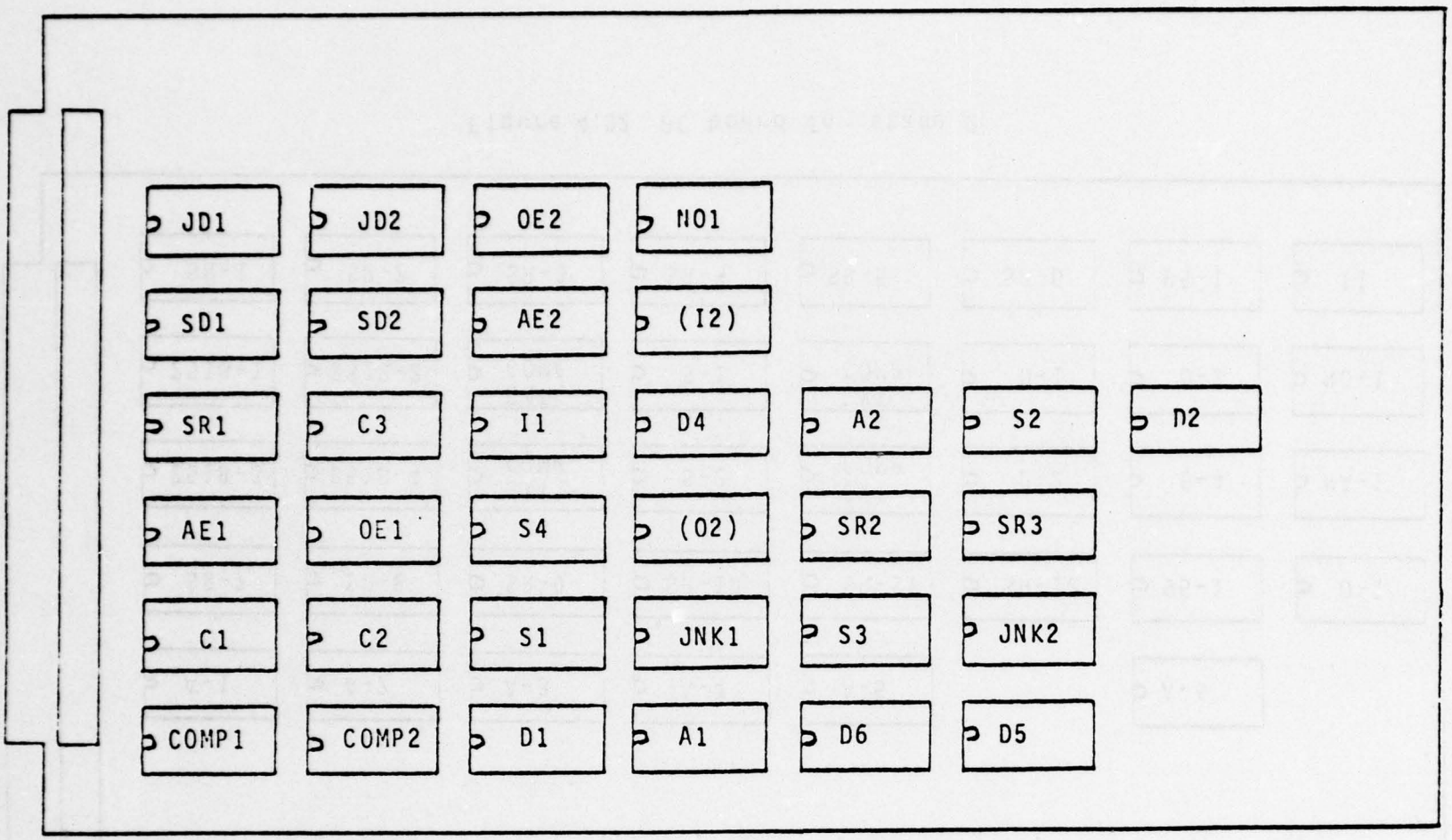


Figure 4.31 PC board for stage 1

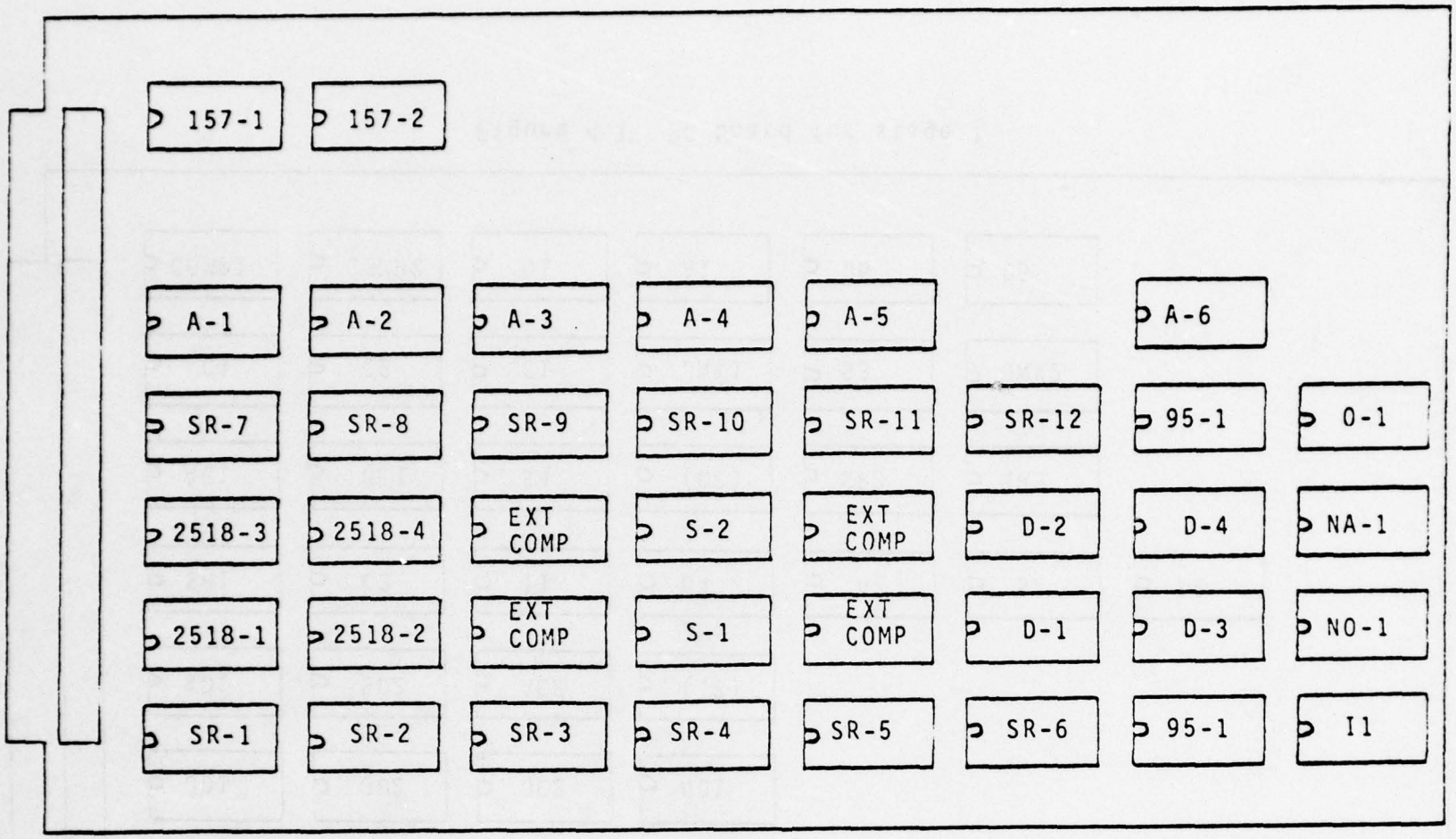


Figure 4.32 PC board for stage 2

Loading VF and DD data

Eight bits of the VF or DD word are parallel loaded into SR-1 when a load pulse is generated by the coder. This loading is accomplished when the load pulse (LD), in conjunction with S3, pulls the load line of SR-1 low. As SR-1 loads only on a clock pulse, the LD pulse enables pin 4 of A1 so that a delayed 3.088 MHz clock pulse can be applied to SR-1 through O1, therefore, permitting loading. The delayed 3.088 MHz pulse is initially delayed by S1 and is further delayed by the serial connection of several unused gates. This extra delay (of approximately 40 nsec) is necessary because the load line of SR-1 must be low for at least 30 nsec before the load clock pulse occurs.

Loading next channel active information

Before a VF or DD data word is loaded, the multiplexer must have stored information concerning the validity of that word. Five (out of the eight between LD pulses) clock pulses after a LD pulse, channel active information concerning the next channel is available. To store this information, therefore, the multiplexer must wait until the CA information is valid and then latch it.

Each LD pulse causes a F-F of D2 to toggle so that $\overline{Q}$ is high. This enables pin two of A3 so that C5 now sees the following clock pulses. After the fifth pulse (because five was previously loaded into C5), the MIN line of C5 goes high. This accomplishes several things. First, in conjunction with a section of I1, the C5 MIN line presets pin 4 of D2 thereby

pulling pin 2 of A3 low. This prevents any further clock pulses from reaching C5. Second, again in conjunction with I1, pin 11 of C5 is pulled low, thereby loading 5 into C5 to prepare it for the next LD pulse. Finally, the MIN line of C5 clocks pin 3 of D3, thus latching the now valid CA data.

Loading and use of L information

As will often be the case, less than the entire 8 bits of a VF word will be used. L, or the number of bits of the data word to use, is available to the multiplexer at the same time the word is loaded (when LD occurs). By being used, it is meant that a particular bit is shifted from SR-1 to one of the output stages. Two conditions can exist at the time L is to be loaded:

1) The previously latched CA information indicates that this channel is not active. In this case, a low was latched for CA and therefore pin 5 of D3 is low. This pulls pin 2 of NA-1 low and pin 3 of NA-1 high. This then forces the load pin (11) of C3 high and the L information is never loaded. C3's MIN line remains high (it went high at the end of the last valid word) and, in conjunction with a section of I1, pulls pin 1 of A1 low. This prevents any clock pulses from reaching SR-1; and, hence, none of the invalid word is shifted out. The next LD pulse will erase this word by loading over it.
2) The previously latched CA information indicates that this channel is active. In this case a high was latched for CA

and, therefore, pin 2 of NA-1 is high. Pin 1 of NA-1 goes high only when the LD pulse is present and the 3.088 MHz clock is low (pins 4,5,6 of A3). An inverted clock is used to prevent any clock slivers from occurring as would happen if C3 were loaded and the MIN line fell while the clock were high. When pins 1 and 2 of NA-1 are both high, NA-1 pin 3 goes low; L is loaded; and C3 counts the following L clock pulses. During this time, MIN is low and in conjunction with a section of I1 pins 1 and 10 of A1 are high. This allows clock pulses (L of them) to be enabled to SR-1 and the output stages. Note, as mentioned before, the clock of SR-1 (the feeding circuit) is delayed with respect to that of the output stage (the receiving stage). After L clock pulses have occurred (L bits have been transferred to the output stage), the MIN line of C3 goes high and disables any further data transfer.

Generation of C193 pulse (start of new frame)

This pulse signals the end of a frame and the start of the outputting of a new frame. D9 from the receive card is inverted by a section of I1. This, then, presets pin 10 of D3 which forces pin 9 of D3 high. This enables pin 12 of A1 so that the next LD pulse appears at pin 11 of A1 as the C193 pulse. C193 inverted is also required and is provided by a section of I1.

Test position marker

Chips C2, C1, Comp 2, and Comp 1 will be removed during regular circuit operation. This section is a position of on-board circuit testing and can be used if a variable position marker is ever required. The desired position is applied in binary to Comp 1 and Comp 2. C1 and C2, which are reset by C193, then count as the frame is being outputted. Pin 6 of Comp 2 will go high every time the desired position is reached. If only one position is to be marked, pin 6 of D1 may be used as the marker. Note, pin 6 of D1 will go high and stay high after the first position is reached.

Loading of signaling, channel assignment and framing bit

Even though that particular section is not being used, the DDR receive card is still generating gating pulses to nonexistent channel cards. One of these (the pulse for Channel 6) is used to parallel load the signaling, channel assignment, and framing bit. This pulse will now be referred to as LLD. When LLD goes low, a section of I2 forces pin 4 of A2 high. This enables a series of load clock pulses to reach SR2 and SR3. A series of pulses occurs instead of a single one due to the length of LLD. As long as the data remains the same, however, the series of load clock pulses will have the same effect as a single pulse. Pins 15 of SR2 and SR3 (the load pins) are brought low during LLD by a section of I1.

Insertion of framing bit, signaling, and channel assignment into the frame

As just explained, this information is loaded during a frame and is held until the start of the next frame. When C193 occurs, 13 is loaded into C3 and the MIN line goes low. This, in conjunction with a section of I1, forces pin 9 of A2 high. This enables the information being shifted out to be put on line. Note, the shift clock to SR2 and SR3 occurs on the falling edge of the 1.544 MHz. This prevents the data from shifting on the rising edge of the 1.544 MHz clock as the output DF-F latched on the clock's rising edge.

C193 also presets pin 10 of D2. This forces pin 8 of D2 low which is passed to stage 2 to disable the parallel output stages. When 13 bits have been used, MIN of C3 goes high. This, in conjunction with a section of I1, pulls pin 9 of A2 low and thus prevents any further shift clock pulses. MIN of C3 also forces pin 5 of NA-1 high. Pin 4 of NA-1 is connected to the pulse that occurs on the falling edge of the 1.544 MHz clock. Therefore, on the falling edge of the thirteenth clock pulse after C193 has occurred, NA-1 pin 6 goes low and clears pin 13 D2. This clearing forces pin 8 of D2 high which enables the output stages of multiplexer stage 2. Note, on the rising edge of the thirteenth clock pulse, the output D F-F latched the last bit of valid information from SR3. On the thirteenth falling edge, the output D F-F was connected to one of the parallel output stages. Therefore, on the fourteenth rising edge, the

D-FF latches the first bit from one of the output stages. By changing the source of data on the falling edge, the output F-F has a continuous output stream.

Stage 2

Components

The types and designation of components used in stage 2 are given in Table 4.13.

Table 4.13 Component identification - stage 2

Designation	Type
A__	74LS08
D__	74LS74
NA__	74LS00
NO__	74LS02
O__	74LS32
S__	74LS123
2518__	7518
6x166	6 74LS166 connected in series
27__	74LS27
95__	74LS95

Stage T/S

The output section has two identical output stages. These stages alternately load and output the last 180 bits of a frame. This switching is accomplished by the C193 pulse clocking pin 3 of D4. The outputs of the F-F pins 5,6 of D4) are then ANDed with the appropriate clocks to determine the stage's function. If the stage is loading, it receives L clock pulses (derived from the 3.088 MHz clock)

per valid data word. If the stage is outputting, it receives the last 180 1.544 MHz clock pulses per frame.

Data transfer

When a stage is in the load mode, the T/S F-F enables the serial data from SR-1 of stage 1 to the input of that output stage. When a stage is in the output mode, high level bits replace the data as it is shifted out. These high bits, if not replaced by valid data during the next load, will become the frame's 1 fill. This is accomplished by pins 1,2,3,4,5, and 6 of A5 and pins 1,2,3,4,5, and 6 of D1.

Output stage configuration

The 180 bits accommodated by the output are divided among two 2518s, six 74LS166s, and a single 74LS95. When such a stage is in the load mode, the fill bits are removed from the input; and the valid data bits from stage 1 are loaded into the parallel 2518s. Each following load forces the data forward in the shift register network. Finally, during the output mode, each register network is given 180 clock pulses. Thus, any valid data is shifted out, and the registers are loaded with fill for the next data load sequence.

4.6 Demultiplexer and signal extraction

The demultiplexer is the last circuit in the DDR receive section. Its function is to create two T1 bit streams and to transmit these PCM streams to the receiving T1 systems. Its inputs are parallel 8-bit PCM words, signaling information, and framing bits.

Input sections

There are three types of input circuits associated with the demultiplexer. The parallel PCM data words are loaded into stages of 74LS95 latches. The signaling information is provided, in serial, by the bit stream decoder to 74LS83 8-bit serial in parallel out shift registers. Finally, the framing bit is loaded by a 2-bit shift register constructed of a 74LS74 D F-F.

Output sections

The output of the demultiplexer consists of a section of 74LS157 2-to-1 line selectors and 74LS74 D F-Fs. The parallel PCM data words are converted to serial through the action of the 74LS83 shift registers. When appropriate, the associated 74LS157 disables the output of PCM words and inserts a framing bit in the serial output bit stream.

T/S of 2101-4 RAMs

In each DDR frame, there are 8 bits of signaling information. These bits must be collected and inserted as the

least significant bit in the demultiplexer's output stream. This insertion must occur every sixth frame. To accomplish this, the demultiplexer time-shares two 2101-4 RAMs. Every twelve frames, the function of each RAM is alternated between loading and outputting signaling information.

Explanation of figures

Figure 4.33 presents an overall schematic of the demultiplexer and signal extractor. The locations of components on the circuit board are given in Figure 4.34. The reader can refer to these drawings as necessary while reading this chapter.

Detailed Configuration

Components

Table 4.14 lists the component designations and component types used in the demultiplexer.

Table 4.14 Component identification - demultiplexer

Designation	Type
A__	74LA08
D__	74LS74
I__	74LS04
NA__	74LS00
NO__	74LS02
O__	74LS32
R__	2101-4
S__	74LS123
95__	74LS95
157__	74LS157
164__	74LS164
166__	74LS166
191__	74LS191

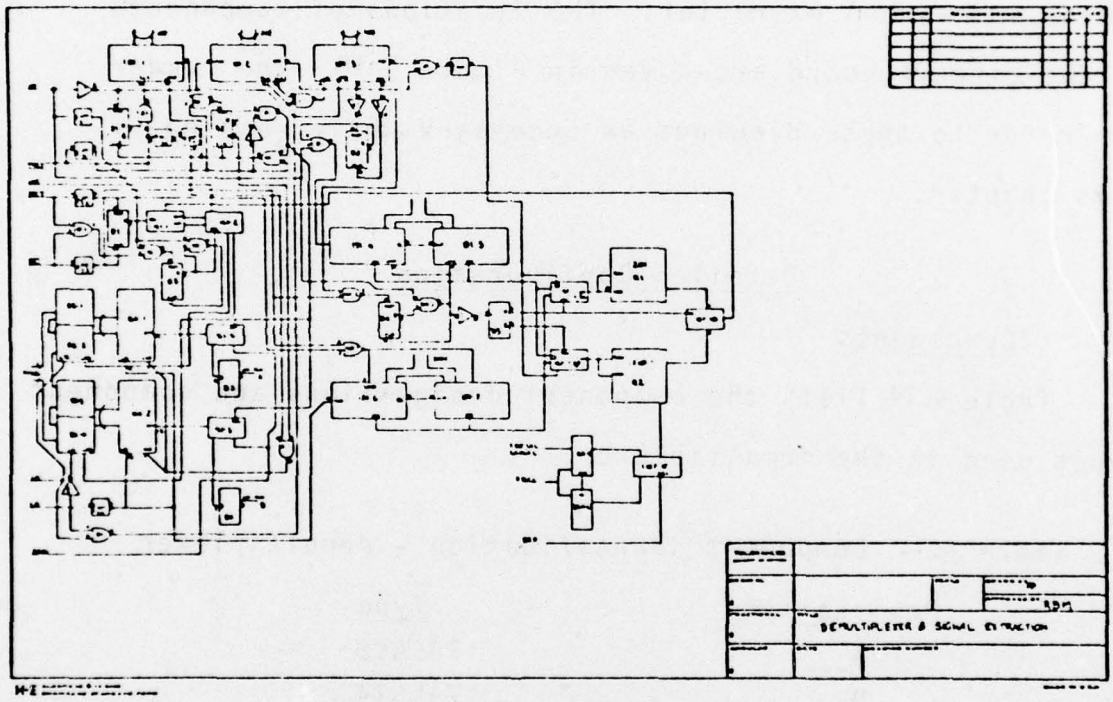


Figure 4.33 Demultiplexer schematic

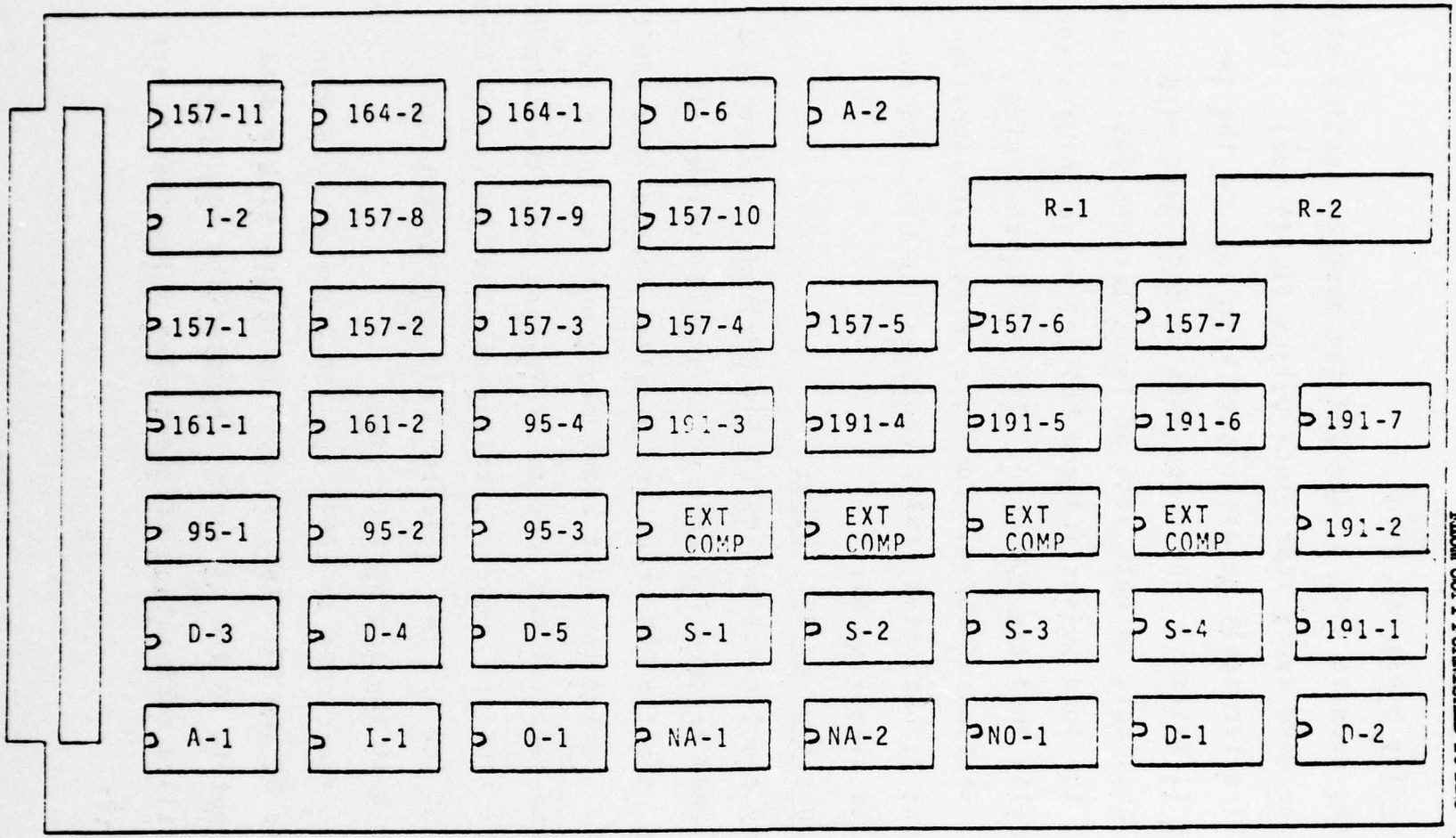


Figure 4.34 PC board for demultiplexer

Loading 8-bit PCM words

In the receive side of the DDR the decoder section will process one word every four clock pulses (1.544 MHz). These words will alternately be associated with each of the two receiving T1 systems. The seven MSBs of each data word are made available to both sets of input latches (95-1,2 and 95-3,4). The load pulses (LA and LB) determine which stage will actually load the PCM word. The LSB is channeled through a section of 157-2. During a frame that signaling information is outputted, 157-2 replaces the LSB with the appropriate signaling bit.

Generation of PCM output

Every eight clock pulses (1.544 MHz) one complete PCM word has been latched for each receiving T1 system. At this time, the two PCM words are parallel loaded into 166-1 and 166-2. During the next eight clock pulses, these 166's convert the parallel PCM word to serial and pass the data to the output 74LS74s. However, this data first passes through sections of 157-3. This selector chip will insert the framing bit at the appropriate time.

Note, the clocks to 166-1 and 166-2 are delayed with respect to D3 and D4. This is done to insure stable data at the input to D3 and D4 during the clocking sequence. For a detailed explanation of the timing delay theory, see section 4.5.

Insertion of the framing bit

In order to reset the demultiplexer every frame, a New Frame (NF) pulse is provided by the decoder. This pulse occurs as the demultiplexer is loading the last word of a frame. A new frame, therefore, will begin exactly eight clock pulses later.

The NF pulse, in conjunction with a part of I1, presets a FF of D1. This enables the following clock pulses to reach 191-1. This 74LS191 has had the number 8 previously loaded. Therefore, on the 8 clock pulse following NF, pin 12 of 191-1 goes high. This presets pin 4 of D2, thus forcing pin 5 of D2 high. This causes the select line of 157-3 to be high. Therefore, on the next clock pulse, a framing bit is inserted in the serial output bit streams. On the following edge of this clock pulse, a section of NA-1 clears pin 1 of D2. This ultimately forces the select line of 157-3 low. Therefore, data is once again enabled to the output 74LS74s.

Transfer of signaling

For every frame the bit stream decoder provides eight bits of serial signaling information. This information is converted to parallel form by 164-1 and 164-2. This is further broken down into a pair of 4-bit signaling words. Chip 157-11 in conjunction with pin 9 of D2 creates the two words. These 4-bit words are then loaded into the T/S RAM network.

RAM T/S

After the thirteenth clock pulse of every frame, the eight bits of signaling information are valid and being held in 164-1 and 164-2. After 40 clock pulses, the minimum line of 191-2 goes high. This causes S4 to generate a write pulse to the RAM T/S network. When another 40 pulses have occurred, the minimum line of 191-3 generates a write pulse to the RAM network. Thus, after 80 clock pulses of every frame, both 4-bit signaling words are loaded.

Counters 191-4 and 191-5 generate the write addresses. These addresses are incremented at the NF pulse and 52 clock pulses into a frame. As the signaling information is updated every 12 frames, the write counters are reset by pulse SIG 1. SIG 1 is a timing pulse, generated by the T1 receive card, that occurs every 12 frames. This reset pulse is also used to toggle a section of D5. When the reset pulse occurs, a RAM has been completely loaded and its function must change to output. This change is accomplished by D5 in conjunction with 157-4,5,6,7,8, and 9.

Counters 191-6 and 191-7 provide the read addresses. These addresses are incremented every time the 8-bit PCM words are latched into 166-1 and 166-2. The counters are reset on the falling edge of SIG 1 and SIG 2. These lines from the T1 receive card go low at the beginning of the frames that are to carry the signaling information. Once again, the addresses are passed to the RAMS by 157-4,5,6, 7,8, and 9.

Note that there are two lines used to indicate a signaling frame. This is done because different signaling information is passed on the sixth and twelfth frames. The correct signaling information for any particular frame is chosen by a section of D1 and 157-1. The selector 157-1 then passes the correct LSB to 157-2 and then to the 74LS95 input stages.

4.7 Bit Stream Decoder

Just as the multiplexer must create a DDR frame from various inputs, the bit stream decoder is required to decompose the incoming DDR frame into its components. These components are the voice data words, dedicated data words, signaling information, and channel assignment bits.

Input sections

All data enters the bit stream decoder in serial. Several 74LS164s convert this data to parallel. The data is then passed, in parallel, to the following stages.

RAM T/S

The circuits in the receive section of the DDR require the data words in parallel. However, the DDR frame is a serial stream of ADPCM data. Enormous timing problems would result if each word were converted to parallel and then immediately passed to the following stages. These parallel words could only be held valid for one clock pulse as the next pulse would indicate the arrival of a new word in the serial bit stream.

With the above timing problem in mind, the bit stream decoder was designed to use a time-shared pair of 82S09 RAMs. These RAMs can be loaded during the one pulse valid time of a word. On alternate frames, a RAM can, then, output the parallel word and hold it valid indefinitely.

Actually, the output words need only be valid for four clock pulses.

On-board system timing

In addition to each board's own locally generated timing, the DDR also uses a system-wide set of timing pulses. These pulses divide the DDR's time among the 48 channels. On the transmit side, these pulses (System Latch 1 and System Latch 2) are generated by the voice switch. On the receive side, System Latch 1 and System Latch 2 are provided by the bit stream decoder.

The bit stream decoder, then, regulates the time used to process each word. In addition to this, the bit stream decoder generates 6-bit addresses. These addresses are used locally to control the output sequence of the 82S09s and are used by other circuits to identify the words as the bit stream decoder outputs them.

Local address generation

The common address, generated by the bit stream decoder, ranges from one to 48 every frame. Also, these addresses are valid for equal numbers of clock pulses. However, there will not always be 48 data words in the DDR frame. Even if 48 channels were always active, each word would not contain the same number of bits. Thus, the addresses must be valid for varying amounts of time.

For the above reasons, the common addresses, while ideal as a control for the output sequence, is of no use during the input sequence. During the input sequence,

a word is shifted in and is loaded into a position in one of the 82S09s. The address applied to the RAM must then change before the next word is ready. These variable-duration addresses are provided by a pair of 74LS161s.

Address selection

There are, then, two sets of addresses available. First, the common addresses range from one to 48. These common addresses are each valid for $\frac{1}{48}$ of a frame. Second, write sequence addresses are valid for varying lengths of time. These addresses range from one to the number of active channels in the current frame. The address applied to a particular RAM is determined by a series of 74LS157s 2-to-1 line decoders. These decoders alternate the type of address applied to a RAM. Therefore, it is these decoders that actually provide the T/S nature of the RAM network.

Output section

The bit stream decoder's output merely consists of the 8-bit parallel data words. These words, like the common addresses, are valid for $\frac{1}{48}$ of a frame. However, several factors prevent the use of the 82S09s as output stages. First, outputs must be time shared. Second, the 82S09 is an open-collector device. Finally, the output of an 82S09 is the inverse of the input.

The first and third factors are removed through the use of a pair of 74LS158s. These chips, like the 74LS157, are 2-to-1 line decoders, but the 74LS158 also inverts

its output. Thus, the 74LS158 does solve the time sharing and inversion problems. The uncommitted collectors of the 82S09s are connected to the +5V supply through 2KΩ pull-up resistors. These resistors are supplied by a pair of pull-up packages

Explanation of figures

The bit stream decoder's overall schematic is given in Figure 4.35. Figure 4.36 provides the component location on the circuit board.

Detailed Configuration

Components

Table 4.15 gives the component designation and component type for each chip used in the bit stream decoder.

Table 4.15 Component identification - bit stream decoder

Designation	Type
A__	74LS08
D__	74LS74
EXT COMP	External Components
I__	74LS04
JK__	7473
NA__	74LS00
NO__	74LS02
PU__	7308 (2KΩ pull-up package)
R__	82S09
S__	74LS123
112-__	74LS112
157-	74LS157
158-	74LS158
161-	74LS161
164-	74LS164

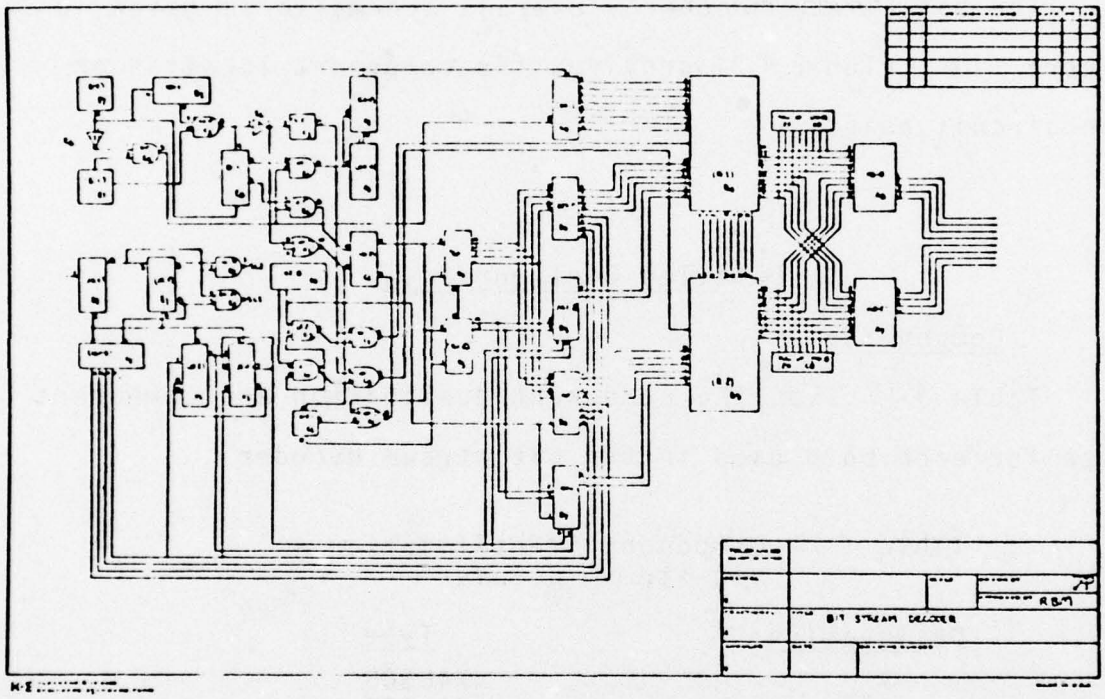


Figure 4.35 BSCD schematic

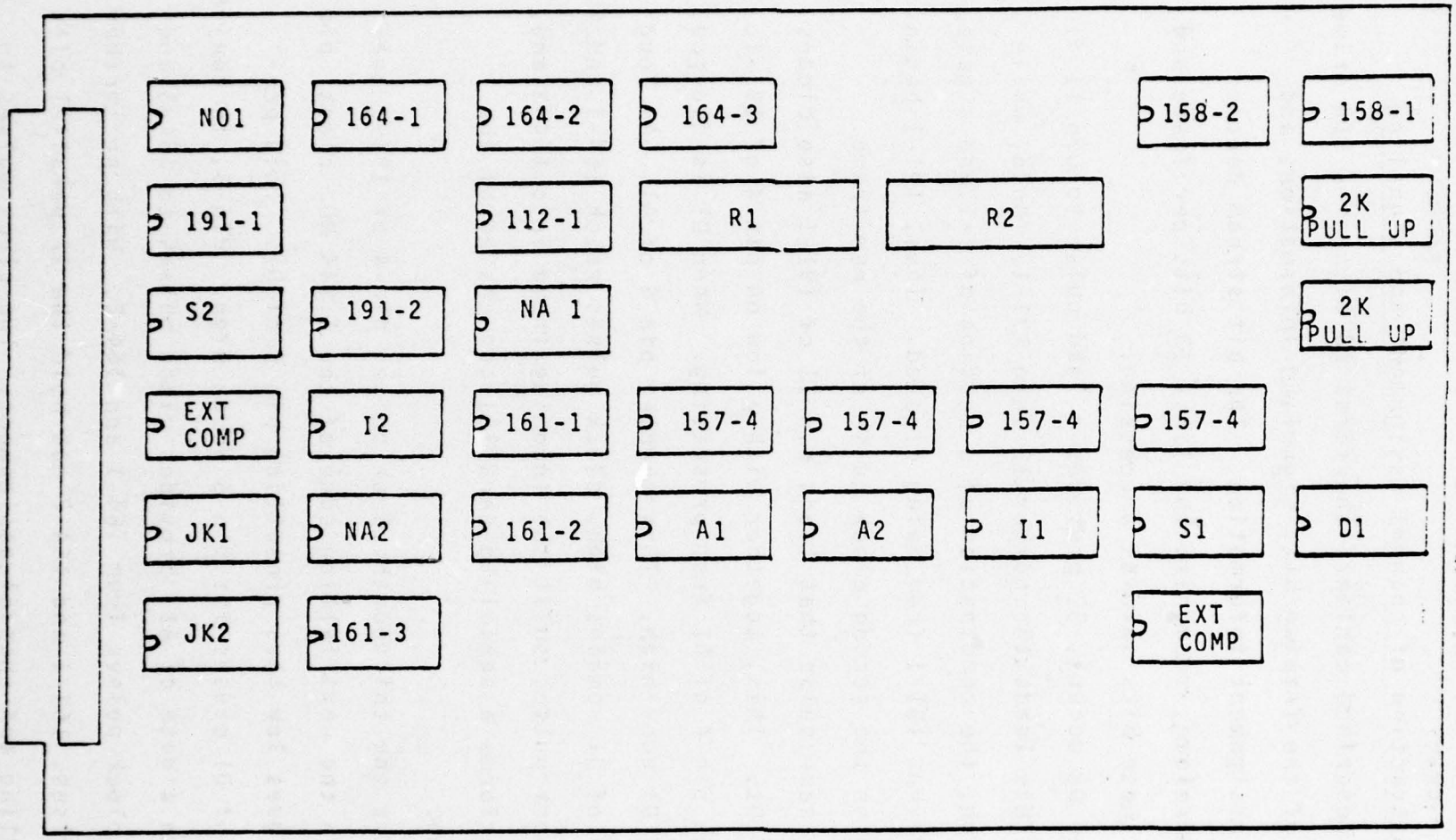


Figure 4.36 BSCD PC board

Extraction of channel assignment and signaling

As described earlier, the first 13 bits of a DDR frame consist of the framing bit, signaling information, and channel assignment information. The bit stream decoder must, therefore, recognize the first 13 bits per frame and provide these bits special processing.

When D9 occurs, S1 provides a load pulse to pin 11 of 191-1. This loads the number 12 into 191-1. Note, while D9 is high, the combination of a section of I-1 and a gate of A1 prevent 191-1 from being clocked. Thus, 191-1 begins to count on the second clock pulse of the new frame.

The same pulse that loads pin 11 of 191-1 also clears pin 1 of D1. This, together with a low on pin 4 of 191-1, will keep pin 4 of D1 from presetting. When D1 is cleared, pin 6 of D1 goes high. This enables pin 4 of A1. As long as pin 6 of D1 remains high, clock pulses reach 164-1 and 2. These clock pulses shift the incoming frame into 164-3 and, thus, performs a serial-to-parallel conversion of the DDR frame.

After the thirteenth clock pulse, pin 4 of 191-1 goes high. On the next falling edge of the 1.544 MHz clock, pin 8 of NA-1 goes low thus presetting pin 4 of D1. This presetting of D1 causes its pin 5 to go high. Pin 5, in conjunction with a gate of A1, provides clock pulses to 164-3 and removes clock pulses from 164-1 and 164-2. With no further clock pulses, 164-1 and 164-2 now hold the 12 parallel bits of signaling and channel assignment. The clock pulses to

164-3 provide for the serial-to-parallel conversion of the data words. The data words will continue to be converted until another D9 pulse begins the cycle again.

Decomposition of DDR frame

After signaling and channel assignment have been removed, the DDR frame consists of fill, voice data, and dedicated data words. A fill inhibit line (connected to pins 2 and 4 of NA-1) is high during the reception of fill bits. With this line high, the 82S09s are disabled, and the fill words are ignored.

With the fill removed, only valid data words remain in the DDR frame. The frame, then, must be broken into these words which vary in length. L (the number of bits in the next serial data word) is loaded into counter 191-2. As each clock pulse occurs, 191-2 decrements by 1, and 164-3 shifts the next data bit in. When the correct number of bits have been converted into a parallel word, pin 12 of 191-2 goes high. This causes S2 to issue a write pulse to the 82S09 that is in the write mode. The correct 82S09 is given the pulse by a combination of gates on A1, A2, and NA-1. The load pulse also increments the address to the write 82S09 and loads the next L into 191-2.

Write sequence

As with any time-sharing scheme, the T/S of the 82S09s is based on a single gate. In this case, the gate is a section of 112-1. This F-F is wired for toggle operation and changes state on the D9 pulse. The Q and $\bar{Q}$ of 112-1 are

then sent to pin 1 of A2 and pin 13 of A1. These connections enable the write pulses to the correct 82S09. Q and $\overline{Q}$ of 112-1 also are connected to the select pins of 157-1,2,3, and 4. In this way, the type of address (read or write) is enabled to each RAM.

The address to a RAM in the write mode need only change after a word is loaded. Therefore, unless there are 48 active channels, the write mode counter will not span its range of 1 to 48. This address generation is provided by the combination of counters 161-1 and 161-2. These counters are reset on the D9 pulse and are incremented every time a word is loaded.

Read sequence

The read sequence is only a scanning of a RAM's contents. This scan just presents each 8-bit word at the bit stream decoder's output. Each word is valid at the output for $\frac{1}{48}$ of a frame.

The section of the bit stream decoder that provides the addresses for the read scan is similar to the address generator contained in the voice switch. The 1.544 MHz clock toggles pin 1 of JK-1. Pin 12 of JK-1 then toggles pin 5 of JK-1. These two sections of JK-1, in conjunction with two gates of NA-2, generate System Latch 2 at pin 3 of NA-2 and System Latch 1 at pin 6 of NA-2. These latch lines provide common timing for the entire receive side of the DDR. The actual read addresses are generated when pin 8 of JK-1

clocks 161-3. This counter coupled with both F-Fs contained on JK-2 generates the six read address lines. As with the write addresses, the read addresses are reset on the D9 pulse.

4.8 Voice Switch

The function of the voice switch is the detection of speech in each of the 48 incoming voice channels of the DDR. The theory of operation is the same as that described in [10]. Briefly, the voice input is rectified and compared with a threshold called TH. If this threshold is exceeded three consecutive times, the voice switch is turned on for about 170 m sec. The rectified voice input is used to establish two thresholds, TL and TH. TL is adjusted such that between 3 1/3% and 5% of the voice samples exceed TL when the voice switch is off. This sets TL just below the noise peaks. TH is set 7 levels above TL so that it rides just above the noise. More details are available in [10].

The available inputs to the voice switch are the 1.544 M Hz clock, the frame reset pulse ($\overline{AD9}$), and the voice data consisting of 8 parallel bits per word. The outputs include the voice switch output, address lines, and several timing pulses. The voice switch output goes "hi" when speech is present on a channel.

A block diagram of the circuit that adjusts the thresholds is shown in Figure 4.37. The channel address lines count from 0 to 47 during each frame allowing 4 clock pulses for each voice data word. The input voice data is rectified by simply dropping the sign bit. First, TL is read from an 8 bit x 48 word RAM. The input is compared with TL for 1200 frames and the number of times TL is equaled or exceeded is accumulated in

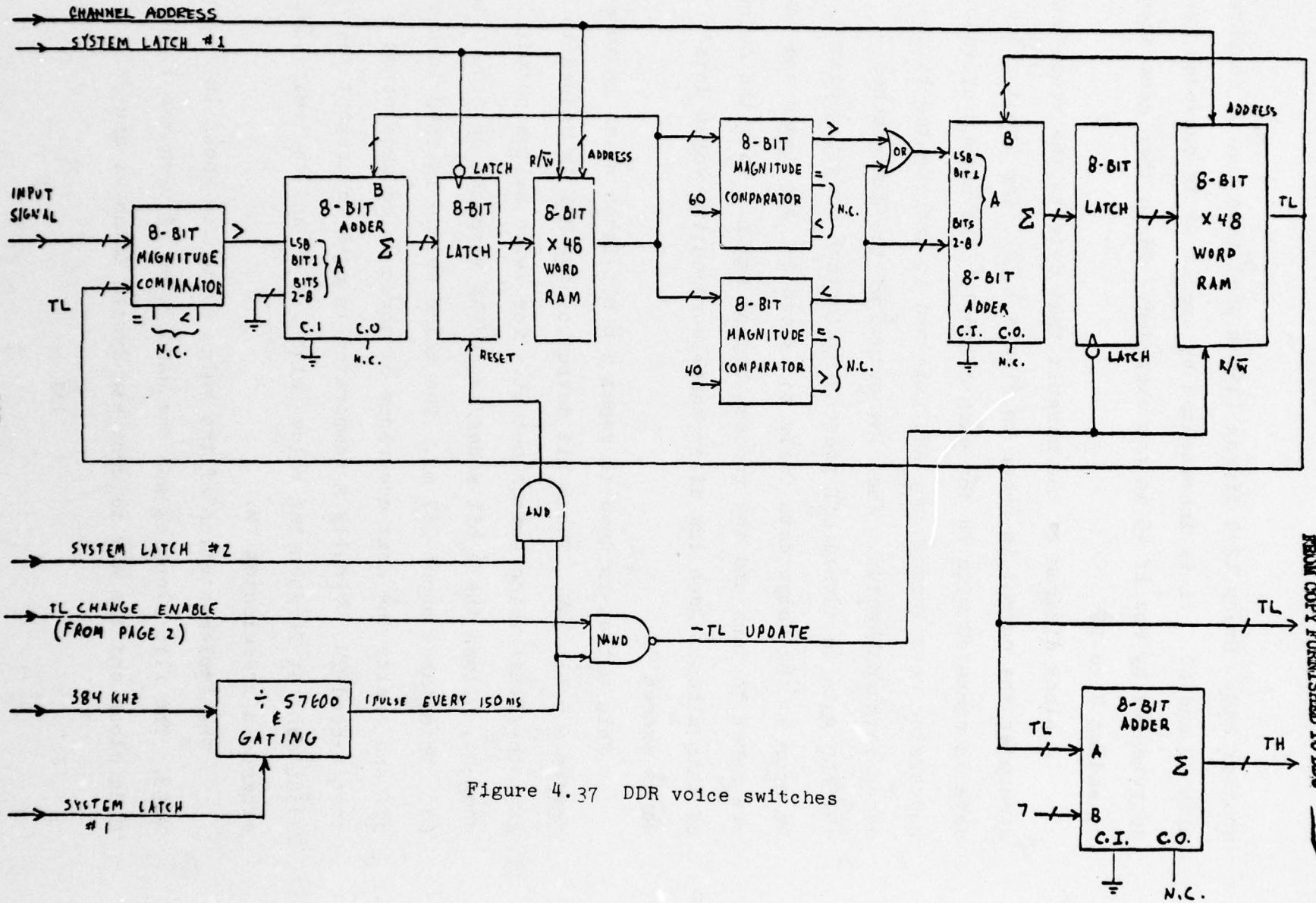


Figure 4.37 DDR voice switches

another RAM. Every 1200 frames (150 m sec) this sum is compared with 40 and 60. TL is incremented by one if 60 is exceeded, and decremented by one if 40 is not exceeded. TH is then generated by adding 7 to TL.

A block diagram of the circuit that detects the voice and generates the output is shown in Figure 4.38. The input voice data is compared with TH for each channel. The output of the comparator goes through the AND gate and is incremented by the adder/latch/RAM setup. When the output of the RAM reaches 3, the AND gate is turned off preventing further counting past 3. As soon as the voice data fails to exceed TH, the latch and RAM are reset to zero and the process begins again. Thus the output of this circuit goes low after three consecutive voice data words exceed TH.

This output is used to reset a 6 bit latch which in turn resets a 6 bit RAM. The 6 bit output of the RAM is NANDed to give the final voice switch output. The voice switch output, when hi, allows the 6 bit adder/latch/RAM to count up from 0 to 63. This takes about 167 ms. The voice switch is then turned off and awaits the next occurance of three voice data words that exceed TH. Finally a comparator is used to prevent the adjustment of TL when the voice switch is on and the voice data exceeds a preselected TM.

The implementation of the voice switch is shown in Figure 4.39. The flip flops c1 and the NAND gates c2 uses the 1.544 M Hz clock and the $\overline{AD9}$ to generate System Latch #1 and #2.

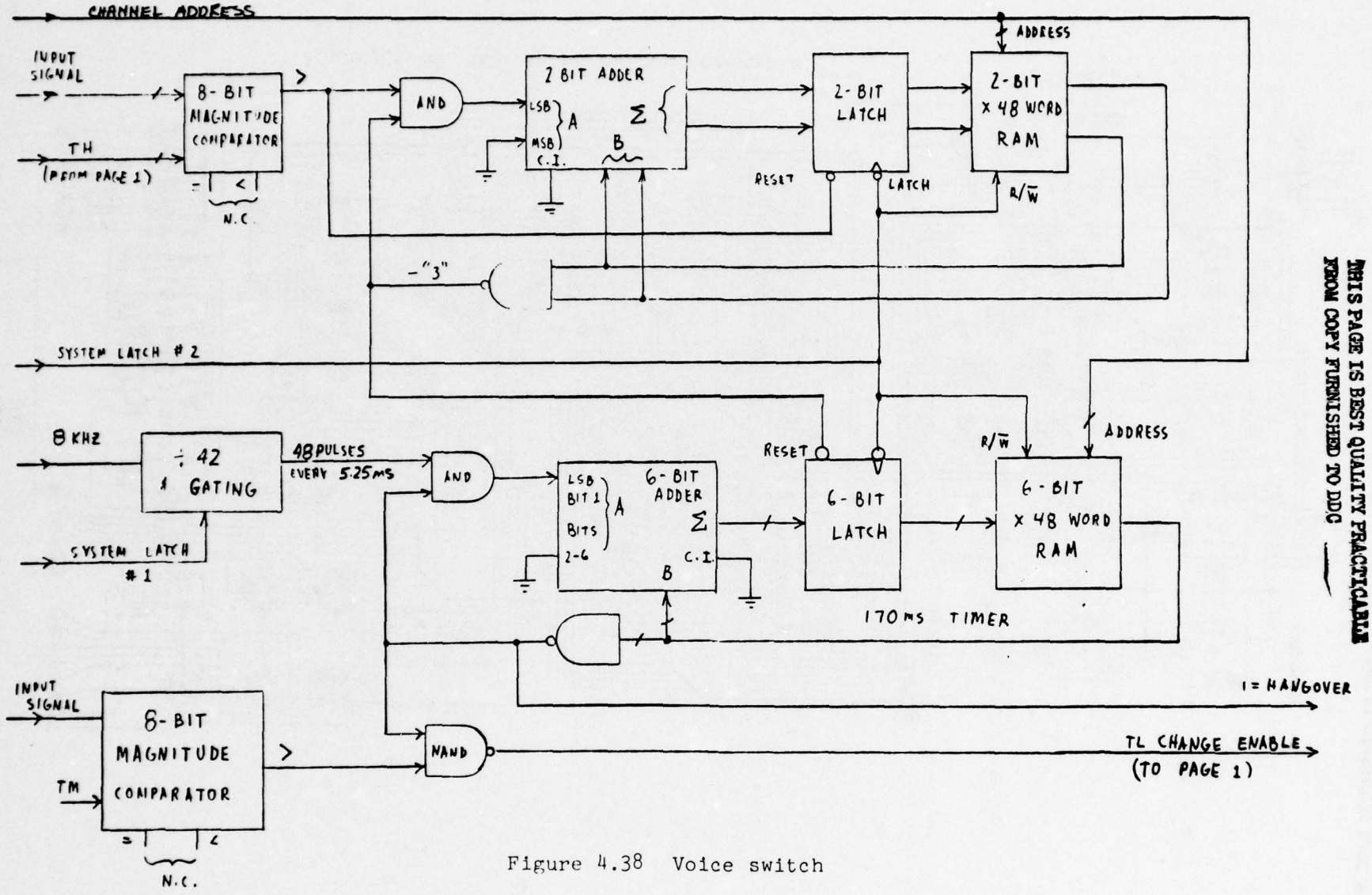


Figure 4.38 Voice switch

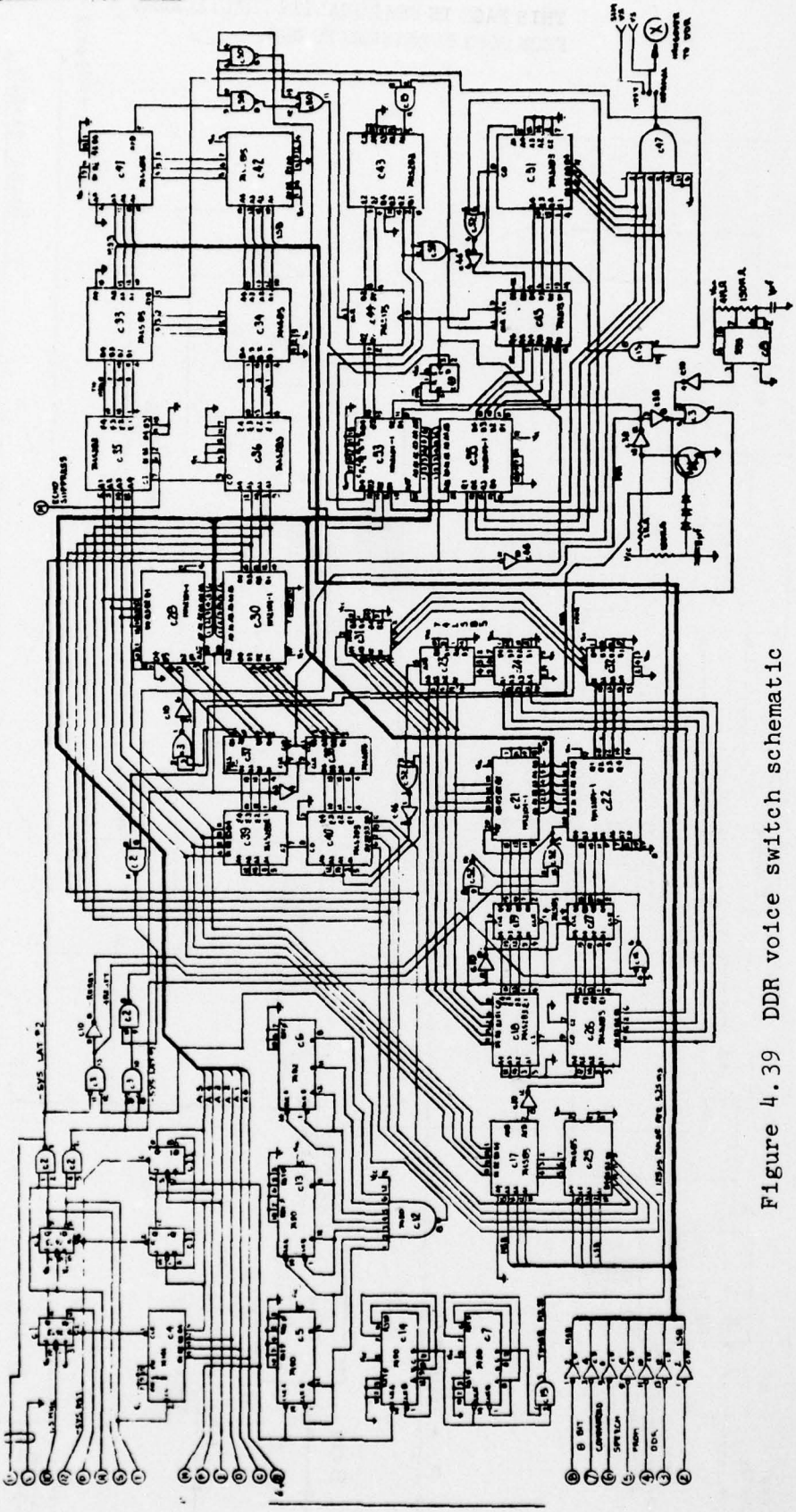


Figure 4.39 DDR voice switch schematic

SL #1 goes low during the second quadrant of each channel while SL #2 goes low during the fourth quadrant of each channel. The counter c4 and flip flops c11 generate the channel addresses. Counters c5, c13, c6, and NAND c12 divide by 1200 and generate a low pulse for one frame every 150 ms. This is NORed with SL #2 to generate + and - Reset. It is also NORed with SL #1 which is NANDed with-disable to get-TL update. Counters c7 and c14 divide by 42 to get a 125 s pulse every 5.25 ms called the Timer Pulse. This completes the timing generation.

Invertors c9 and c10 are used to invert the incoming 7 bit voice word since it is received inverted. Comparators c17 and c25 compare the voice input with TL for that channel from ROMS c28 and c30. If the voice data is not less than TL ($Vd \geq TL$), the adder/latch/RAM c18, 26, 19, 27, 21, and 22 is incremented. This circuit counts the number of times the voice word exceeds TL during each 1200 frame period. At the end of this 1200 frame period (150 ms), the comparators c7, c8, c31, and c32 compare the number of times TL was exceeded with 40 and 60. If the number was less than 40, the adder/latch/RAM c39, c40, c37, c38, c28, and c30 decrements TL by one. If the number was greater than 60, TL is incremented by one. TH is then generated by adders c35 and c36 by adding 7 to TL. Allowance has also been made to add an additional 48 to suppress echos if required. This signal must come from the local receiver.

During power up, the initial TL is set to 64 using NOR c3 and NOT c10 to prevent noisy channels from locking on.

Latches c37 and c38 are cleared using the power up pulse. The NOR c52 and NOT c46 are used to add one to TL or add all ones to TL thus effectively subtracting one. Once every 1200 frames, latches c19 and c27 are cleared by Reset. The two NOR gates of c52 allow the RAMs c21 and c22 to write when + Reset is high or during the second quadrant of each channel (SL #1). However, if the adders c18 and c26 overflow, writing during SL #1 is inhibited. This concludes the threshold level adjusting circuitry.

Comparators c33 and c34 compare the incomming voice data with TH. If the voice data is less than TH, the AND gate c15 is off and the latch c44 is cleared. If hi, the AND gate goes hi and the Adder/Latch/RAM increments. When the RAM reaches 3, NAND gate c50-1's output goes low inhibiting further input from the comparator until a low pulse clears the latch. This low pulse also clears the latch c45 in the Adder/latch/RAM c51, c45, c53, c55. This clears the RAM c55 and turns the voice switch on. The voice switch stays on until the Adder/latch/RAM has counted 31 timer pulses each 5.25 ms apart or about 160 msec. Each time the latch c45 is cleared, the counting is restarted. NOR gate c52 and NOT gate c46 are used to replace a one bit adder. When the voice switch is off, AND gate c15-3 inhibits counting and forces the flip flop c49 to a zero. When the voice switch is on, and if TM is exceeded based upon comparators c41 + c42, the flip flop is set high and disables the TL adaptation until the voice switch turns off again. NAND gate c2 lets $\overline{\text{PPOR}}$ enable the adaptation once every ten seconds to

prevent noisy channels from becoming stuck on. This concludes the description of the voice detection and disable circuit.

Figure 4.40 is the basic timing diagram for the voice switch. The relationship between the $\overline{CLK}$, $A\overline{D9}$, SL1, SL2, and the Address lines is evident. Figure 4.41 shows the Timer Pulse relative to $A\overline{D9}$. The Threshold Level Update pulse is also shown relative to $A\overline{D9}$. Next the Power up timing is shown. The VCC waveform is the 5 volt power supply system. Next, the output of a Darlington pair and a RC network is shown. When applied to the input of NOT gate c46-6 the next waveform, called $\overline{POR}$, is generated. This pulse clears latches c37 and c38 and resets flip flop c49. The $\overline{POR}$ signal is inverted by c46-4 and is shown as POR. This signal is used to set the initial TL level to 64 by forcing the 2nd most significant bit of the RAM c28 to a one. The $\overline{PPOR}$ is a periodic signal that first enables the updating during power up and later prevents noisy channels from becoming stuck on by allowing an update once every 10 seconds. This pulse is generated using a oneshot and NOR c3.

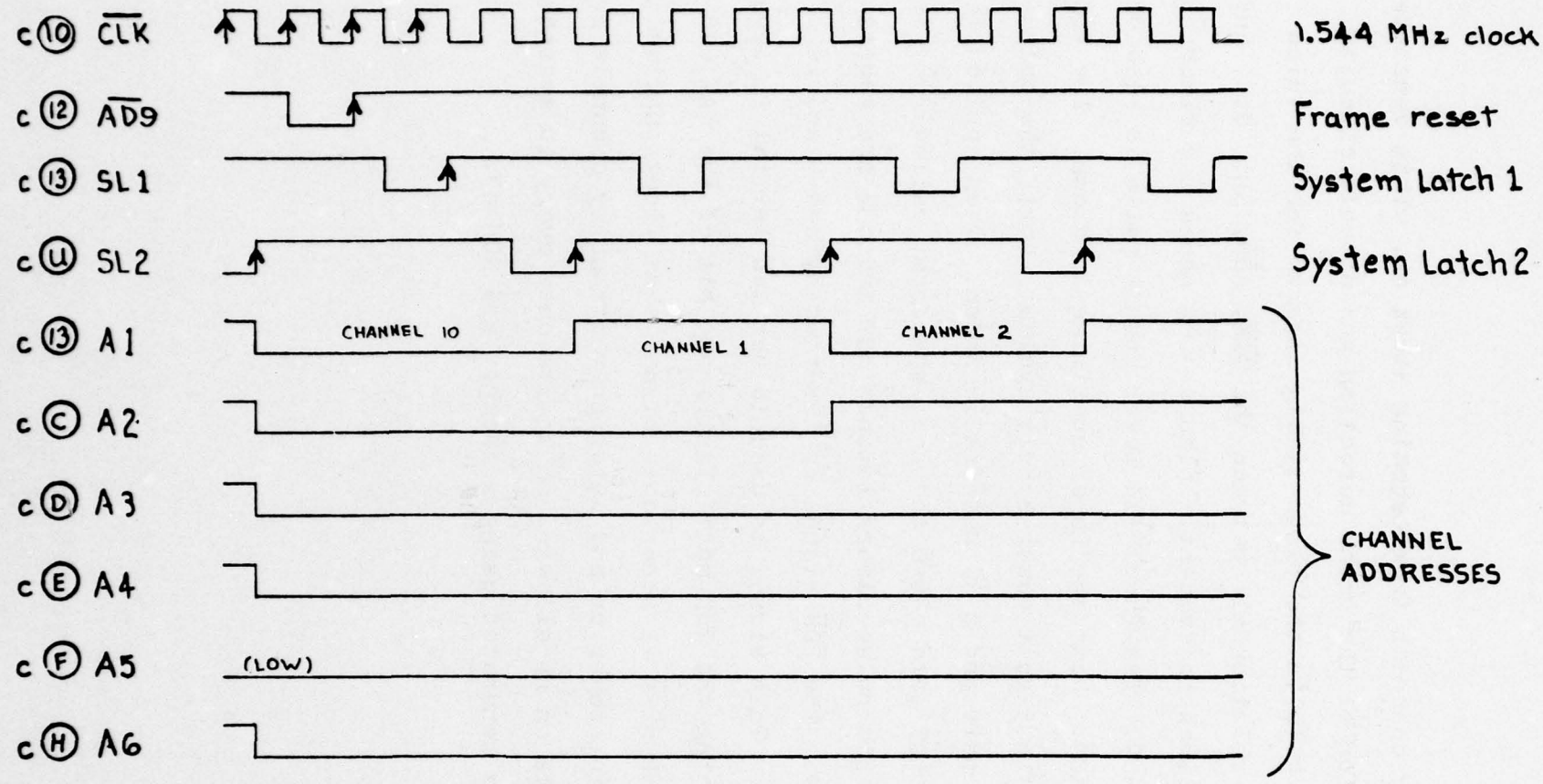


Figure 4.40 Timing diagram for voice switch

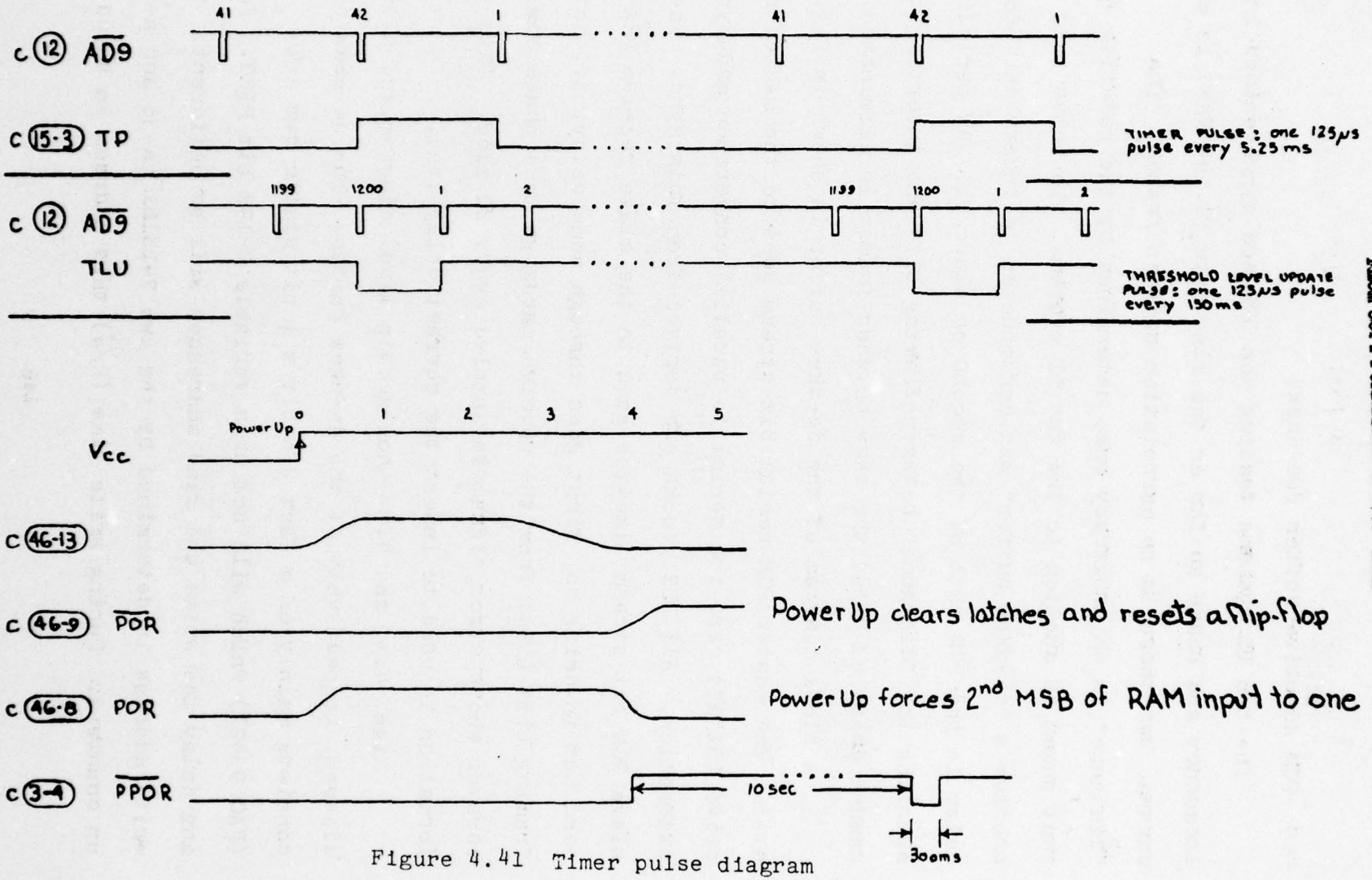


Figure 4.41 Timer pulse diagram

4.9 PCM Adaptive Buffer (De-Skew)

The two T1 systems feeding the DDR are synchronized in frequency and phase as far as the line rate (1.544 MHz) is concerned, but there is no correlation between frames. The "distance" is an arbitrary one, determined by the relative times that power is applied to the two T1 systems. To solve this problem a "de-skew buffer" was constructed to insert the proper delay in the PCM path of the second or slave T1. By periodically sampling the "distance" between framing bits and inserting the needed delay (Δ bits) the skew between frames is eliminated.

A block diagram of the de-skew buffer is shown in Fig. 4.42. The master PCM serial bit stream goes to the master receive (RX"A") card for serial to parallel conversion and sync extraction. All DDR clocks are derived from this sync. The slave PCM bit stream likewise goes to the slave receive (RX"B") card but in doing so, first goes through some delay, Δ. Both framing bits (D9s) from the receive cards go to a phase comparator whose error signal is sampled every 20 msec. This information is used to insert the correct delay ($\pm\Delta$).

Figs. 4.43 and 4.44 show a chip level diagram and a timing diagram, respectively of the de-skew buffer. The de-skew buffer consists mainly of a fast 256 bit x 1 bit static ram a-46 (FAC 93421) which will act as a variable 0-192 bit FIFO. For any delay the write and read addresses will be different. The write address is determined by the two 74LS161 (a-36 and a-37) up counters. During write time ($R/\overline{W}$) this address is added by

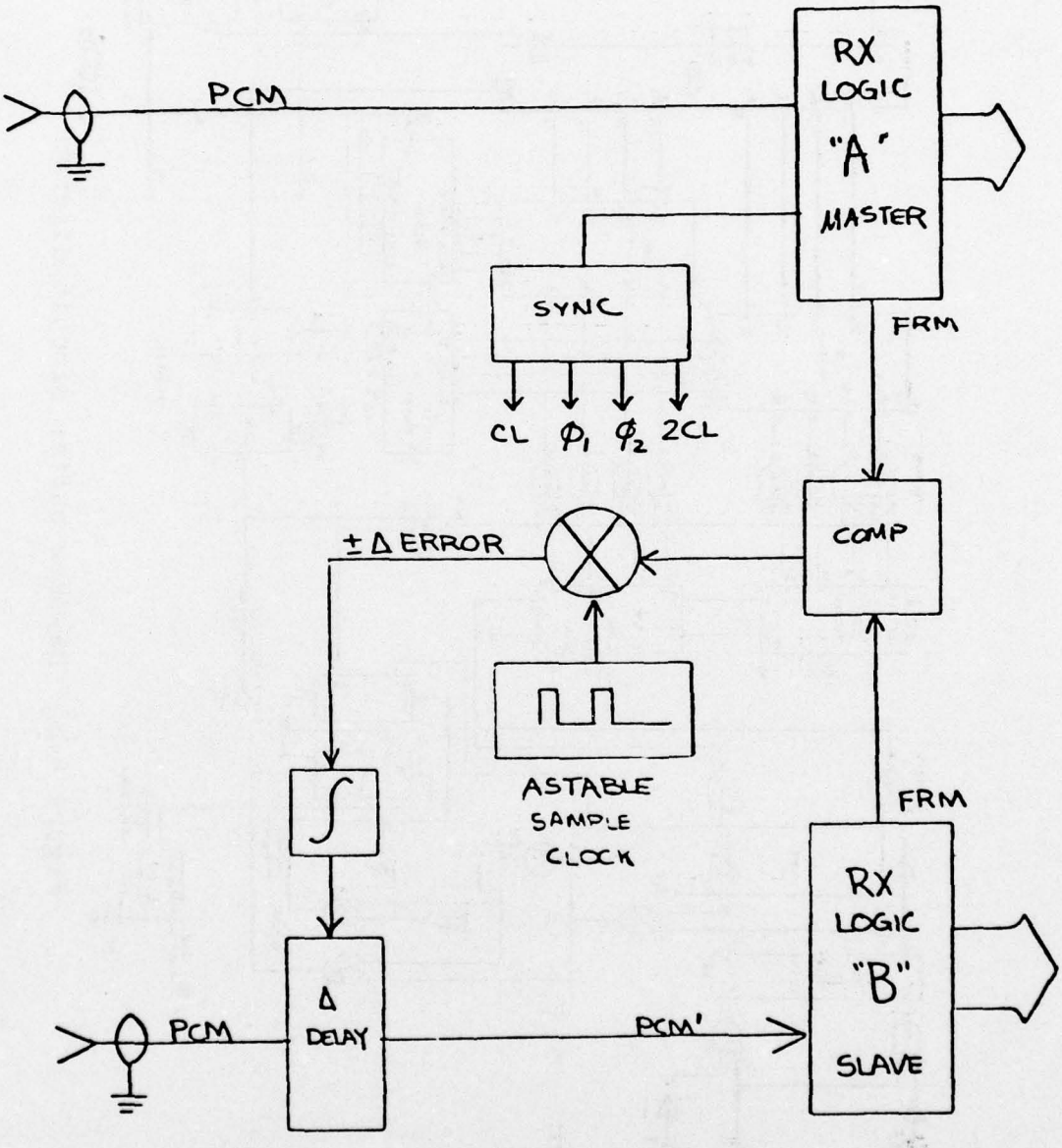


Figure 4.42 De-skew buffer

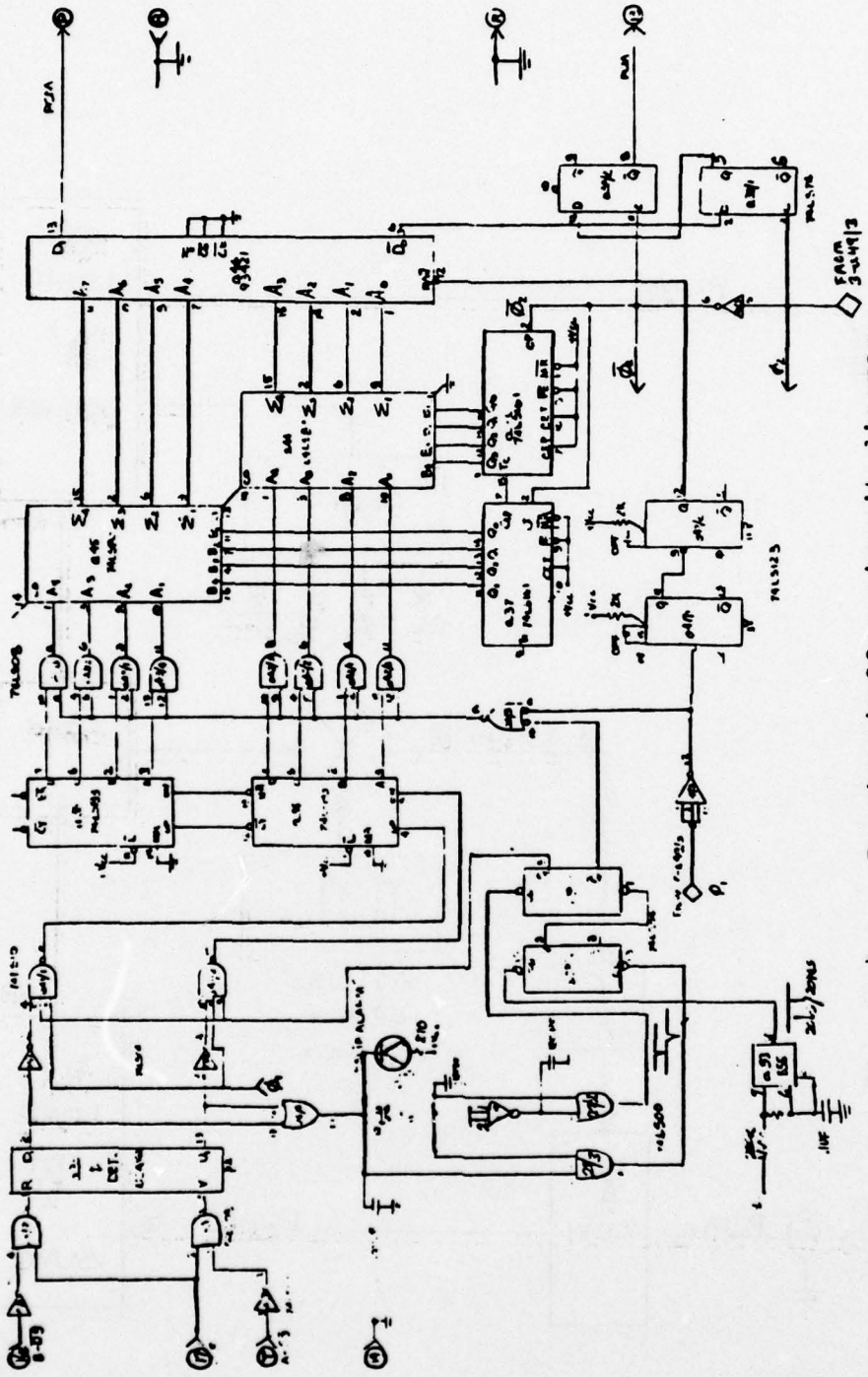


Figure 4.43 De-skew buffer circuit diagram

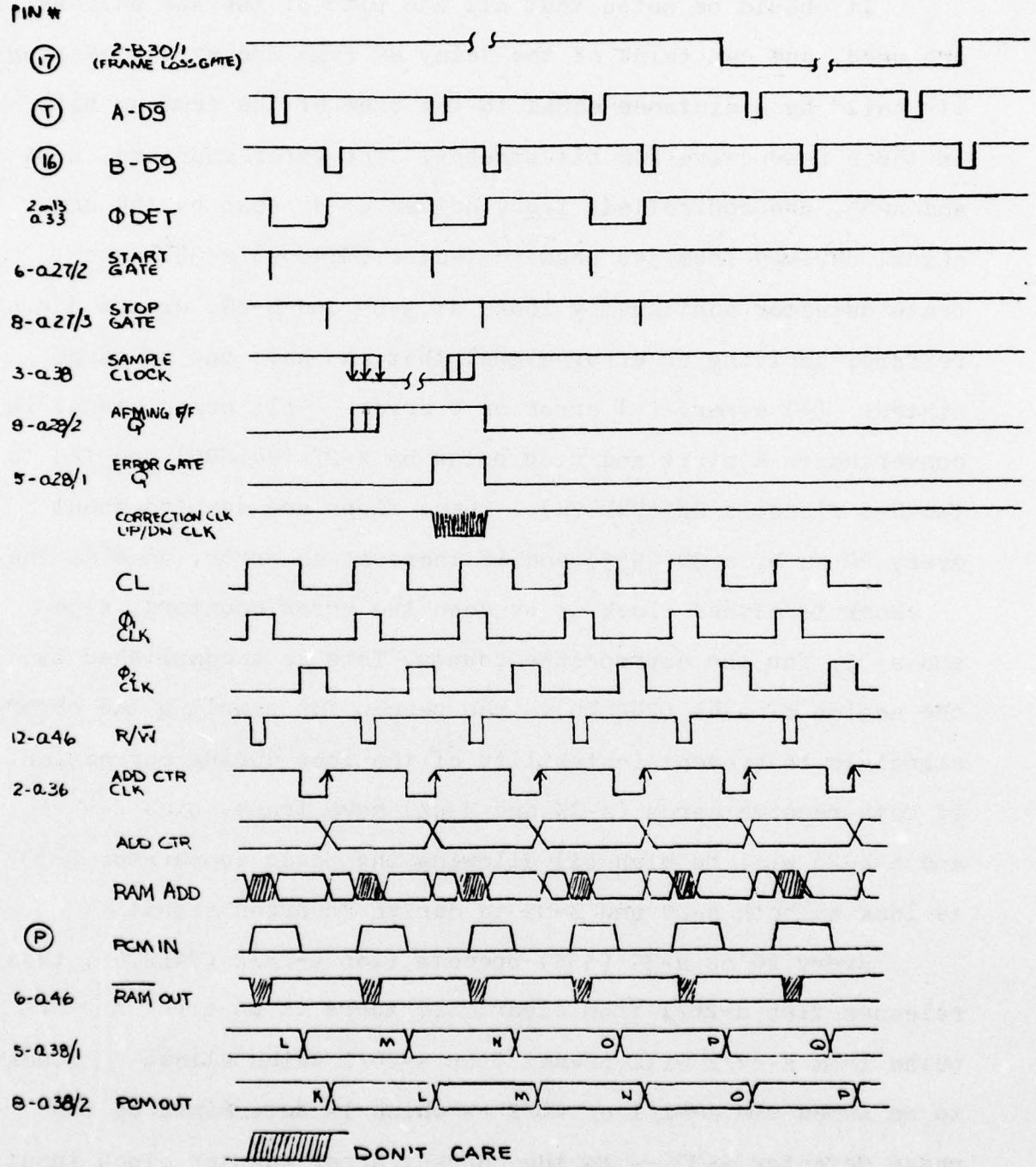


Figure 4.44 De-skew buffer timing diagram

a-44 and a-45 (74LS83) to the address put out by the up/dn error counters, a-34 and a-35 (74LS193) for the correct write address.

It should be noted that all 256 bits of the RAM address are used, one can think of the delay as read and write "chasing its tail" by a distance equal to the skew of the framing bits in their respective PCM bit streams. The error counters, a-34 and a-35, are controlled, i.e., driven up or down by the error signal derived from the phase detector (MC4044) a-33. The phase detector continually looks at A-D9 and B-D9, except during reframe, deriving an error signal that can have one of three states: (+) error, (-) error or 0 error. This error signal is converted to a start and stop pulse by a-27 (74LS00) and fed to two R-S flops, a-28 (74LS74). These flops are sampled about every 20 ms by a-39 (555) and if there is an error, enables the clock to either clock up or down the error counters, a-34 and a-35, for the appropriate counts. This is accomplished by the action of a-41 (74LS10). The reason for sampling the error signal is to prevent instability of the loop during correction. If both receive cards (A-RX and B-RX) have frame, pins 2-a-26 and 5-a-26 will be high (1) allowing the phase comparator 2-33 to look at both A-D9 and B-D9 to derive an error signal.

Every 20 ms a-39 (555) presets flop a-28/2 (74LS74); this releases flop a-28/1 from clear. If there is an error a start pulse from a-27/2 will preset flop a-28/1 which allows clock to be ANDed via a-41/1 or 41/2 -- which is determined by the phase detector a-33 -- to the correct error counter clock input

(up or down) a-34 and a-35. While the error counter is counting its outputs are held off by $\overline{Q}$ of a-28/1 via a-26/3, the 20 ms sampling rate allows the slave receive card (B-RX) to reframe after correction and stabilize before the next sample is taken. Any residual error will be eliminated by the ensuing samples. During the time of any error an LED on the board called the "slip-alarm" will light. Normally, alignment of both frames occurs in about two sample times ($\sim$40 ms) after system turn-on. The two "one-shots" a-47/1 and 47/2 are used to put the write ($R/\overline{W}$) strobe in the "clear" (no race with ϕ_1). D flops a-38/1 and a-38/2 are used to invert the RAM output ($\overline{D}_0$) and clock it in time with ϕ_2. Notice in the timing diagram (Fig. 4.40) that there is skew between $\overline{RAM}$ out and PCM out, which is equal to 2 bits of RAM, so the total RAM length can be thought of as 258 bits.

4.10 Signal Extraction

Signaling is available at the receive card at all times, i.e., Signal Highway #1 and #2 output signal bits at the appropriate channel times (this is true for both A-RX and B-RX cards). The receive card registers are updated every #6 frame and #12 frame time. It is our desire to accumulate all 96 signal bits (2 per channel x 48) and deliver them to the output multiplexer card in groups of 8 bits where they will be output in blocks of 8 bits per frame. Figs. 4.41 and 4.42 show the signal extractor circuit and a timing diagram for the signal extractor, respectively. During frame #12 we sample (a18) all 96 signal bits and store them in buffer register a-11/1, while at the same time, we output 8-bit parallel groups (a-19) to the output multiplexer cards R+L. There are two clocks, one is a burst of 96 pulses which occurs only every frame #12 time during D2 and D3 times of each channel. The second clock is a continuously occurring burst of 8 pulses which occur during D5 through D8 times every channel and frame time. During frame #12 the signal bit selector (a-18) is caused to scan all four signal highway buses each channel time by the action of counter a-10 which is clocked by the burst of 96 pulses. The 96 signal bits are clocked in and subsequently clocked out of the buffer register a-11/1 by a composite of both clocks (3 - a-20/1). The signaling data is delivered in 8-bit parallel words by the serial to parallel converter a-19. The "D" flop a-2/1 is used with a-9 to develop the 96 burst clock.

4.11 PCM Delay

In the DDR it is desirable to delay the PCM data going to the ADPCM encoder. This delay reduces clipping, at the "front end" of the speech passage, by the voice switch. The PCM delay consists of a 4096x8 bit mos RAM memory with read/write addressing such that any number of words delay (up to 4096) may be selected.

Figures 4.45-4.51 diagram the operation of the PCM delay. The read address is derived from the summing of the 12-bit up counter string b48, 47 and 46. With the hexadecimal switches b1, 9 and 17, by adders b38, 39 and 40. During write the switch input to the adders is zeroed by ADDZ at b54, 55 and 56 (78LS08). This delay addressing scheme is, in essence, chasing its tail backwards around the 4096 bit "circle"; i.e., the bit delay is equal to 4096 — $(R_{add}-W_{add})+1$. The +1 is caused by the (74LS175) latch b33 and 41 delay. Since the TMS4060 MOS RAM is dynamic, the chip enable CE is required to cycle from 0 to +12 volts. This is accomplished by clock driver b53 running at twice the write clock speed (see drawing). The two one-shots, b37/1 and 37/2 are used to put the R/$\overline{W}$ pulse "in the clear".

The combiner portion of this circuit is made up to two 74LS157 selectors b25 and b26. They alternately look at both T_1-A and T_1-B receive card outputs during one channel time. This data (8-bit words) is furnished to the delay RAM and to the undelayed latched b34 and b42. The one-shot b44 is used to strobe the latches for correct voice switch timing.

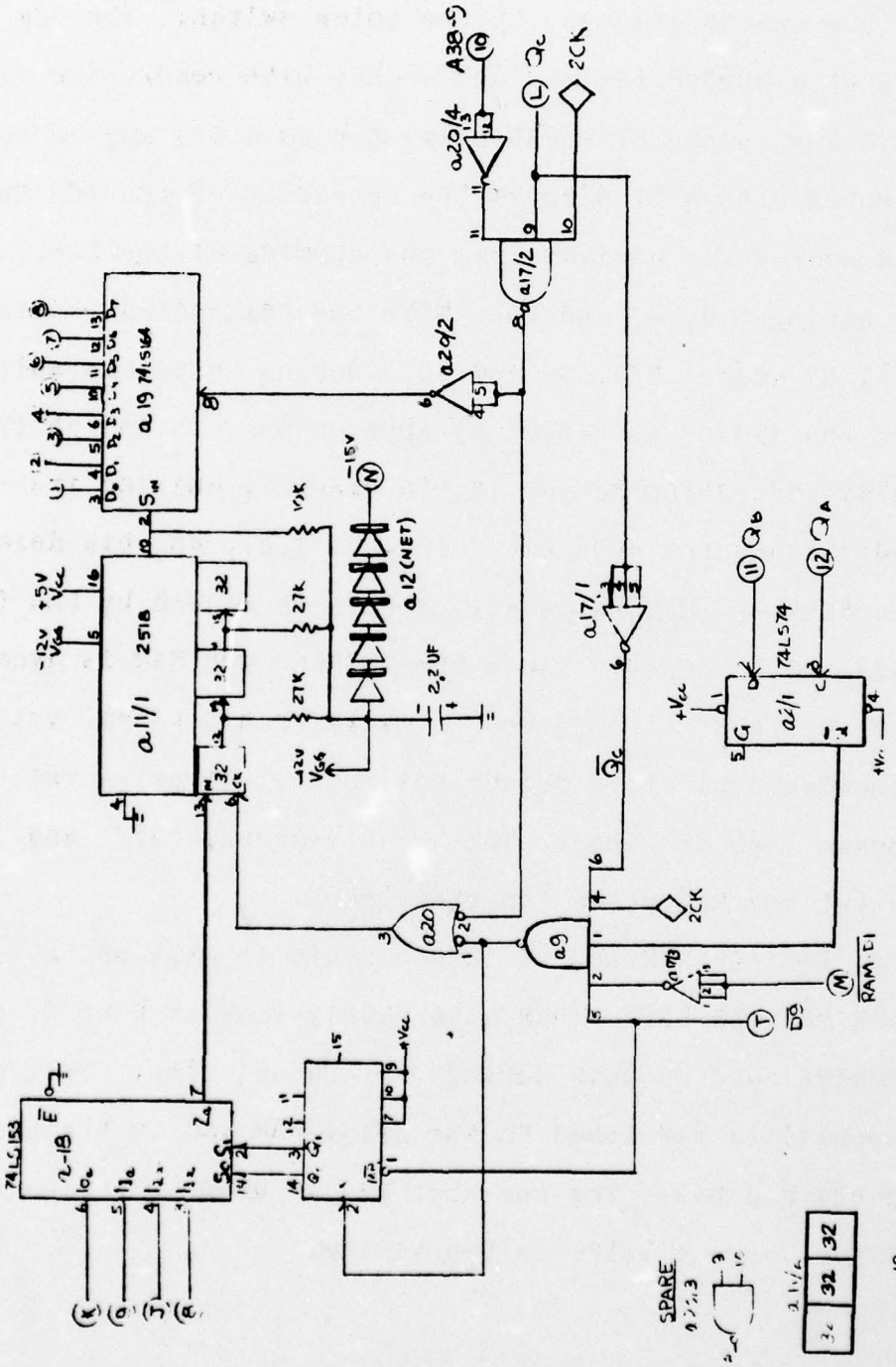


Figure 4.45 Signaling extractor

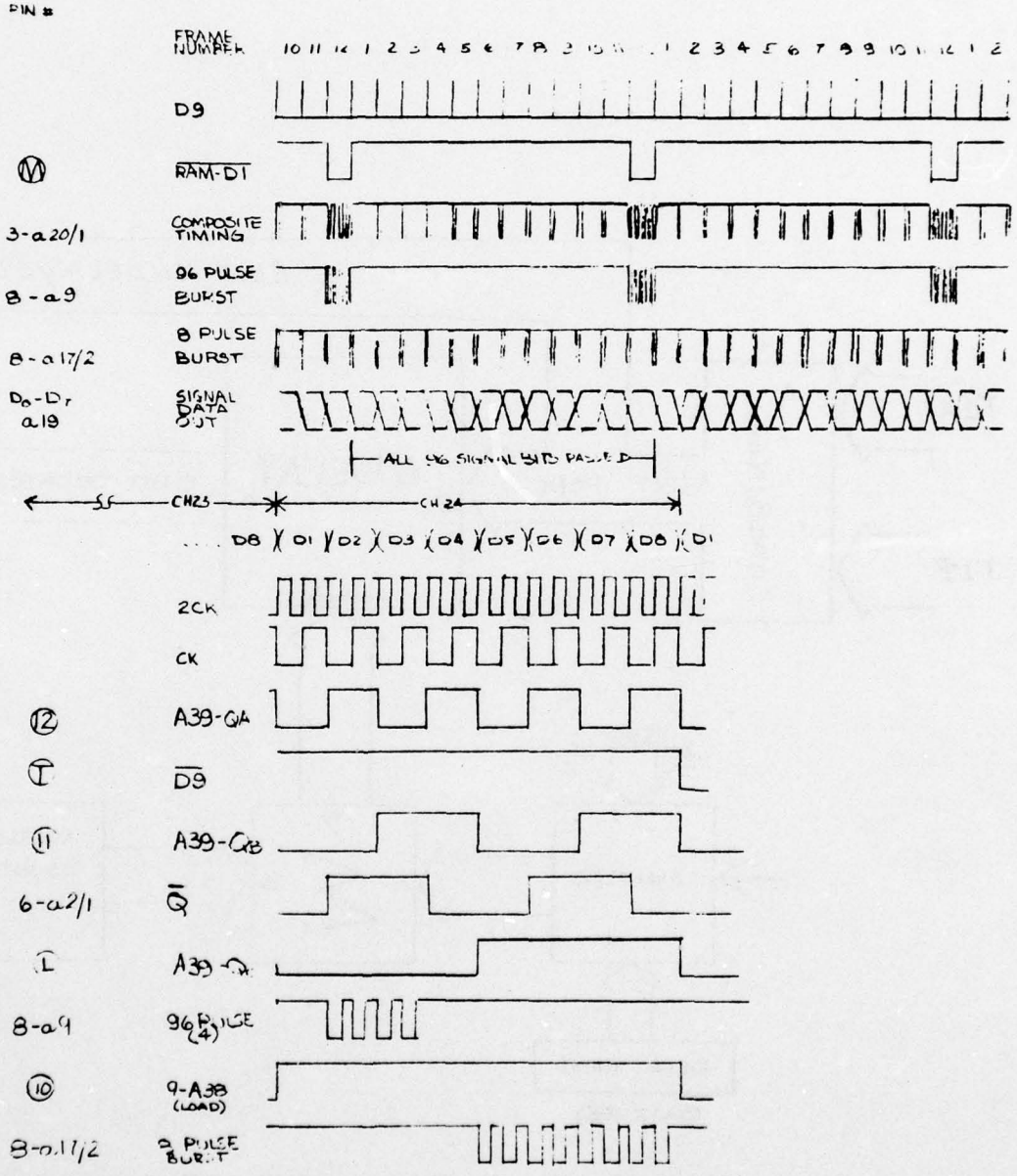


Figure 4.46 Signaling extractor timing

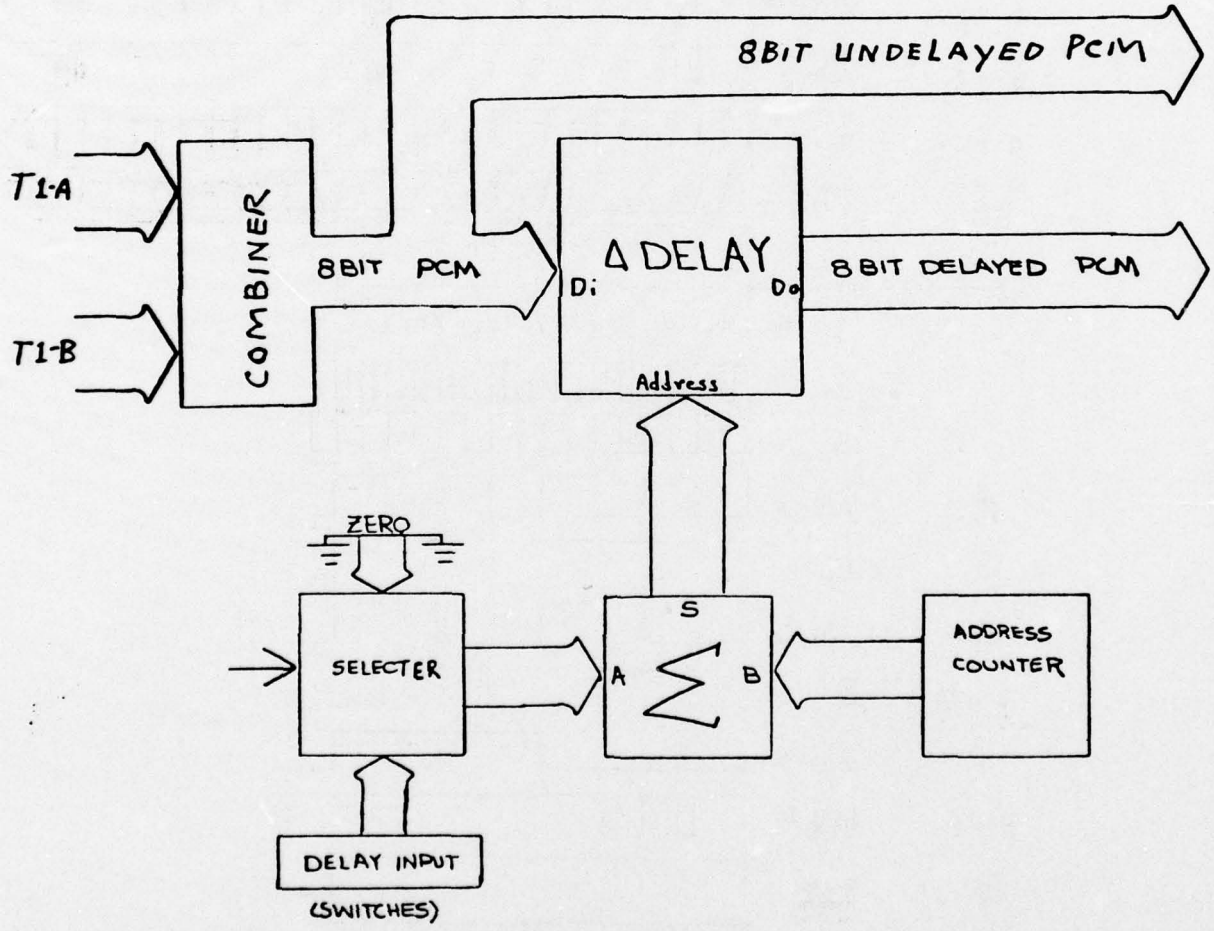


Figure 4.47 Functional diagram of the PCM delay

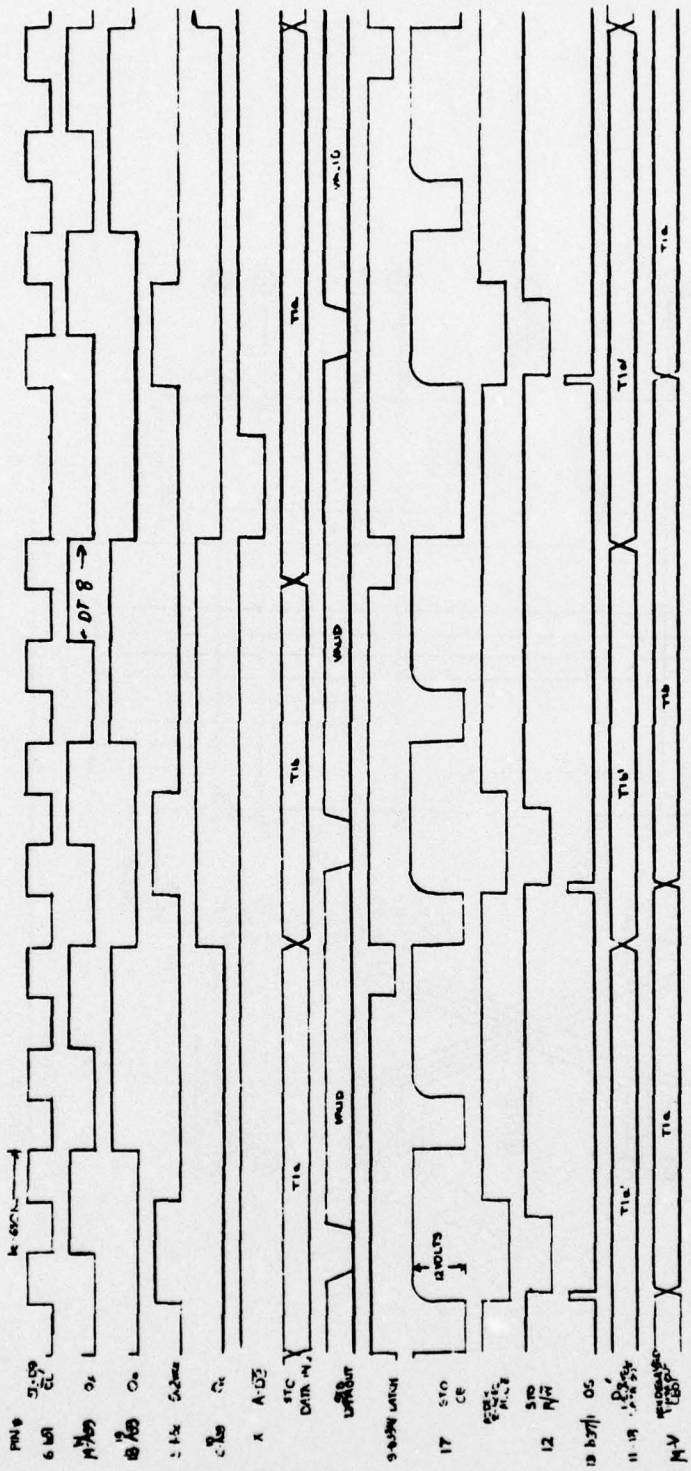


Figure 4.48 Timing diagram of PCM delay

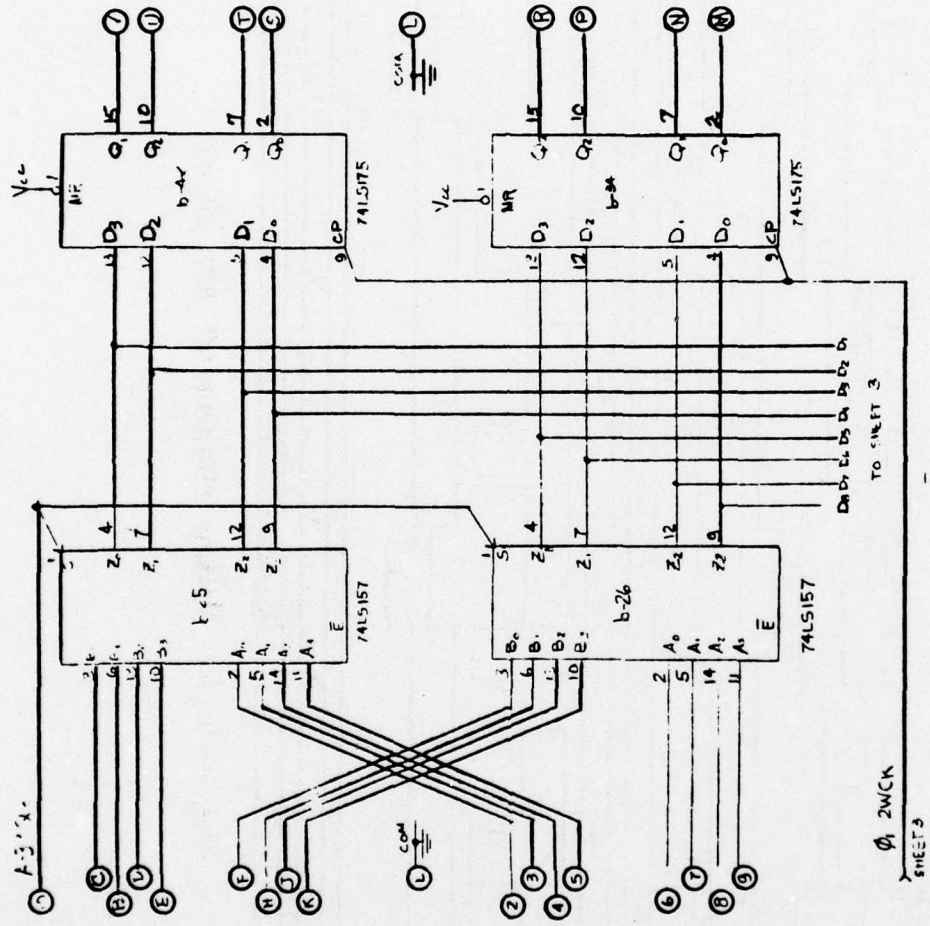


Figure 4.49 Circuit drawing #1 for PCM combiner/delay buffer

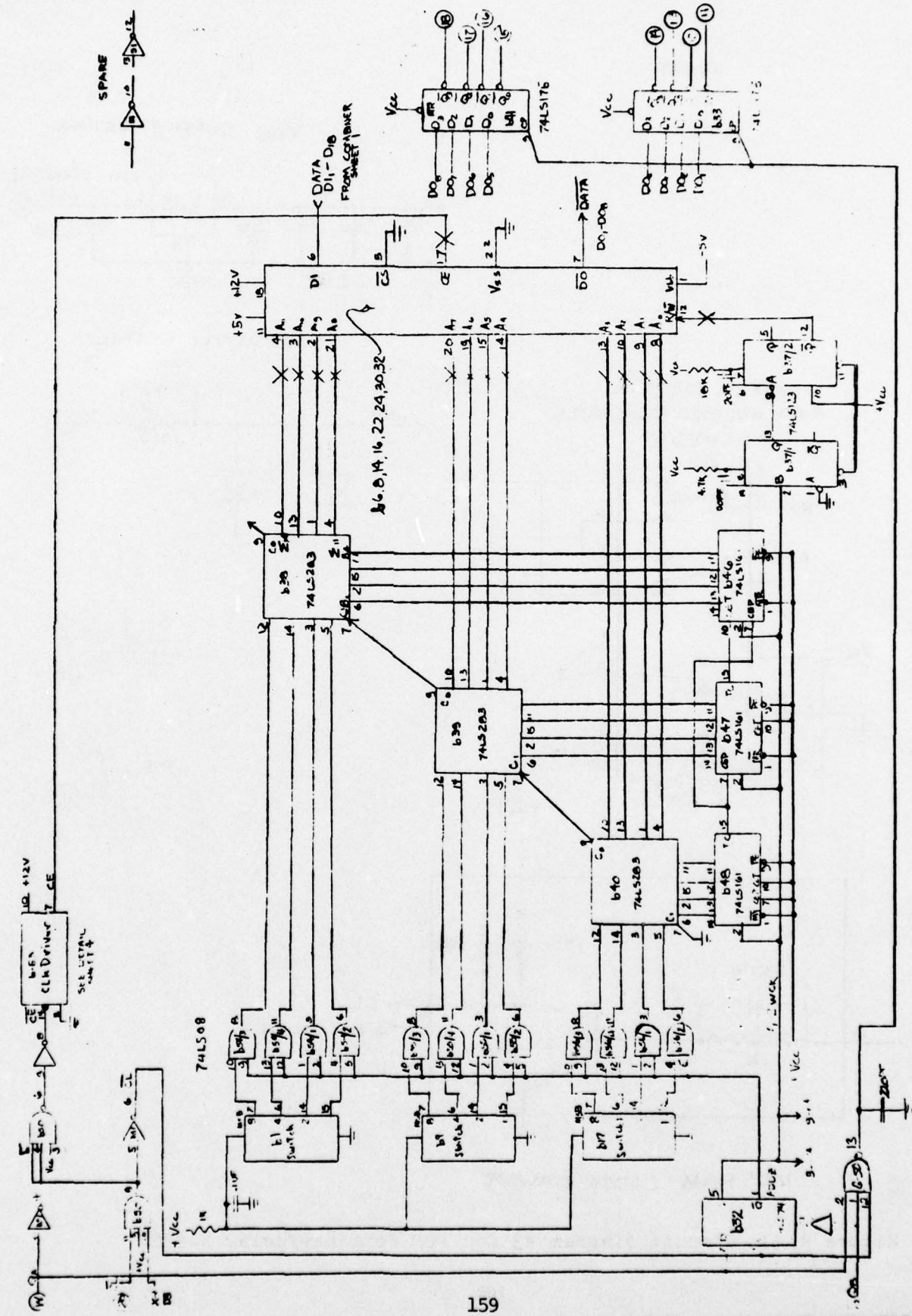


Figure 4.50 Circuit diagram #2 for PCM combiner/delay buffer

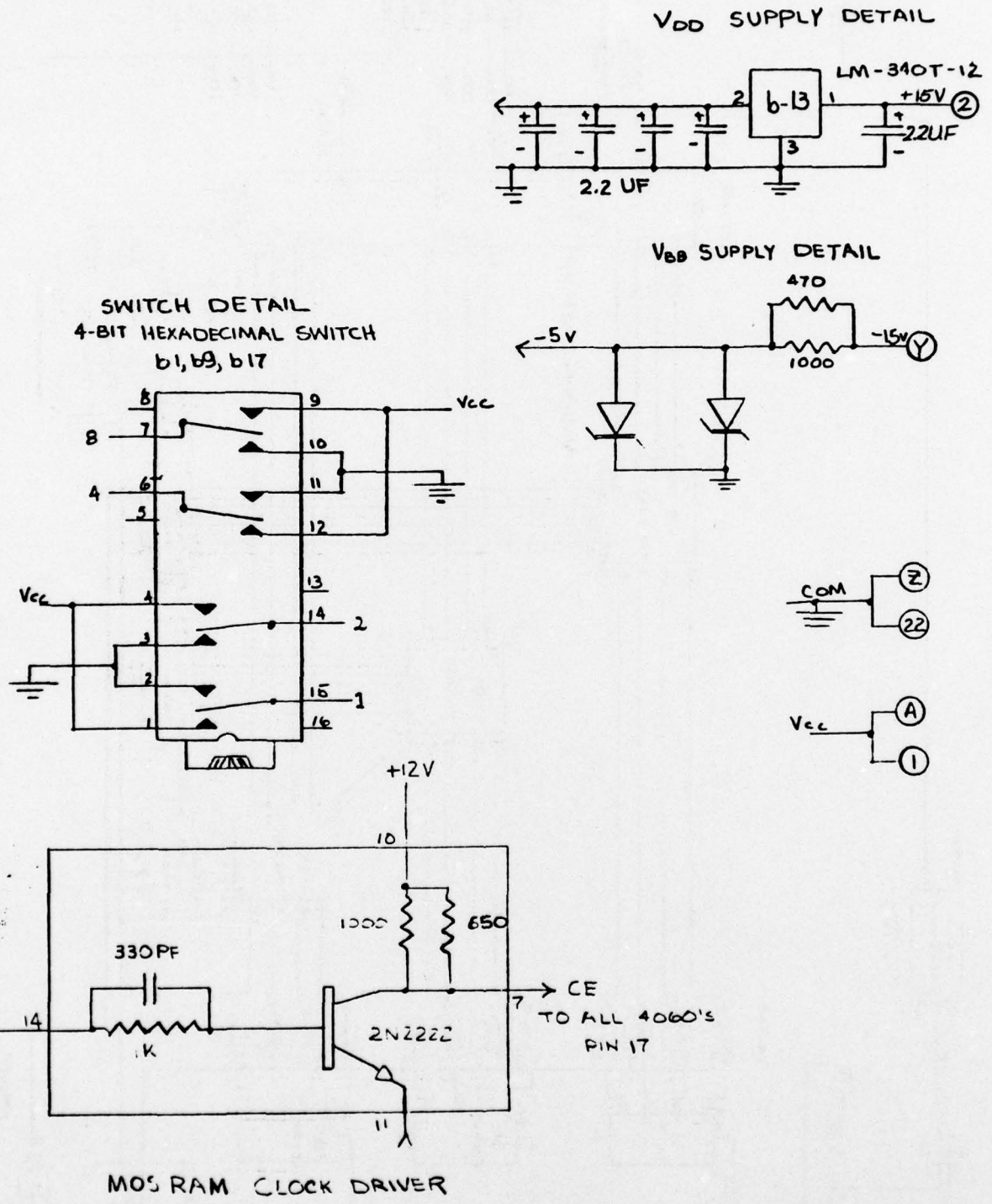


Figure 4.51 Circuit diagram #3 for PCM combiner/delay buffer

4.12 Transmitter Compute L Board (Board j)

The purpose of the transmitter Compute L board is to compute the number of bits to be allocated to each active channel. The card also formats the results of the computation for output to the multiplexer and predictive encoder cards.

Computation Section

It is the job of the computation section to determine the number of bits that will be allocated to each active voice channel. Appendix D lists the number of bits that will be assigned to the active voice channels as a function of the number of active voice channels and the number of data channels.

Notice that if the number of data and voice channels combined exceeds 22, a certain number of the voice channels receive from 3 to 7 bits per channel and the remaining voice channels receive 1 plus this number of bits (although sometimes all of the active channels receive the same number of bits). This scheme insures that all of the available bits in a frame are used.

The following definitions and equations describe the algorithm used to compute the number of bits assigned to each channel.

Input and Output Variables

NACH = Number of Active voice CHannels for next frame to be processed by the multiplexer and predictive encoder cards (obtained from channel assignment).

NDC = Number of Data Channels (obtained from thumbwheel switch on Compute L board).

L = Number of bits allocated to first NL (see below) active voice channels.

NL = Number of active voice channels receiving L bits per channel (remaining (NACH - NL) channels receive (L + 1) bits per channel).

Internal Variables

BITSA = 180 - (NDC * 8) --- Total number of BITS Available to the active voice channels.

ACC = Active Channel Counter

BITC = BITS available Counter

LI = Value of L Internal

Equations used by Compute L

L = MIN (8, BITSA/NACH)

NL = Remainder of BITSA/NACH division (or equal to NACH if remainder is zero).

Fig. 4.52 is a block diagram which illustrates the structure of the transmit Compute L card. During D9, BITC (counters J-42 and J-50) are loaded with -BITSA (1'S complement). This value is obtained from the thumbwheel switch and the 4 bit adder J-43. Also ACC (counters J-19 and J-27) is loaded with - NACH (1'S complement) and LI (counter J-11) is set to zero.

After this preloading sequence the following occurs. On each negative transition of $\phi 2$, BITC and ACC are indexed by one. On the following negative transition of $\phi 1$ if ACC is not equal to -0 nothing

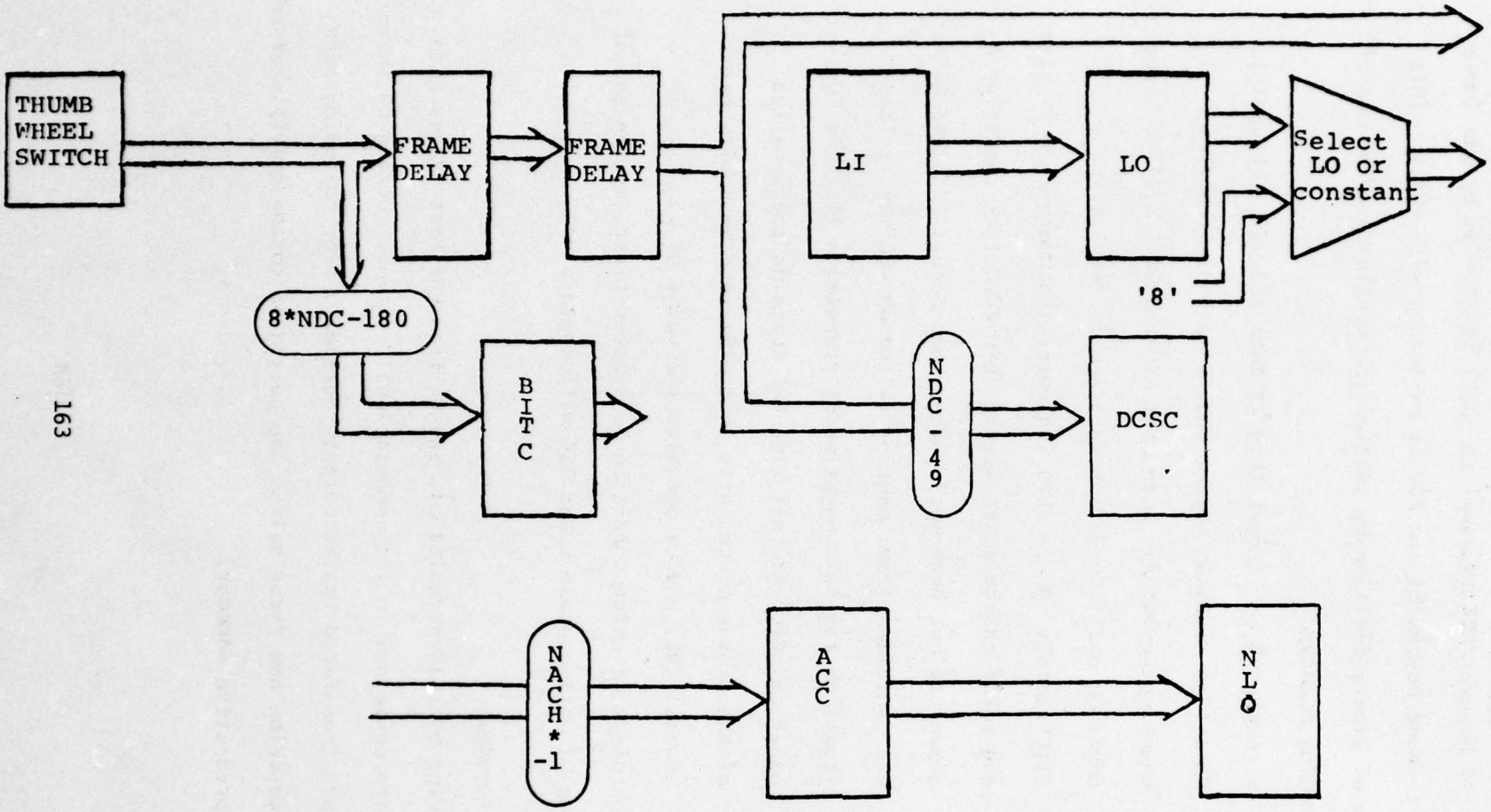


Figure 4.52 Major data paths for transmit compute L board

occurs. If however ACC is equal to -0,LI is indexed by one (note LI cannot count beyond 8) and ACC is re-loaded with -NACH. This sequence of binary division by subtraction continues until one of the following occurs:

A) L reaches 8 --- (fewer than 22 data and voice channels). No action is taken except that L is inhibited from counting beyond 8. Note that ACC and BITC will continue counting until condition B) below is reached.

B) BITC reaches -0 --- This indicates that more than 22 data and voice channels are active (provided that condition A above has not been met). When this occurs BITC and ACC are inhibited from counting (although ACC may re-load itself on the next negative $\phi 1$ transition if it is currently equal to -0). At this time ACC contains the negative of the number of channels that will receive L bits per channel (NL) and LI contains the value of L.

Regardless of which condition is reached first, by the end of the frame LI will contain L and ACC will contain -NL.

Output Section

During D9 the variables LI, and NL from the last frames computation are contained in the computational section. During D9 these values are transfered to the output registers. The following paragraphs describe how these values are passed on to the multiplexer and to the predictive encoder.

The output stages for this section are loaded by the negative transition of D9 anded with ¢2. The counters loaded by this condition are LO (L Output -- counter J-10) and NLO (Number at L Output -- counters J-18 and J-26). D9 also sets a D-flop in package J-49 which allows the loading of DCSC (Data Channel Start Counter -- Counters J-33 and J-41). Note also that the number of data channels selected by the thumbwheel switch is delayed by one frame (this is currently unnecessary but is necessary if at some time in the future the setting of this thumbwheel switch will control the number of data channels on the recieve Compute L card).

The basic clock signal for this section is generated by the D-flops in package J-44 and is called CAS (Channel Acquisition Strobe). There are 48 CAS pulses in a frame. The CAS pulses are 1 slow clock long and are 4 slow clock pulses apart.

When CAS is valid, a signal called ACT (Active Channel) is supplied to the card by channel assignment. ACT is high to indicate that the channel slot being "looked at" by the multiplexer and predictive encoder is to be occupied by an active voice channel.

During CAS if ACT is low nothing occurs. If, however, ACT is high the following occurs. If D-FLOP J-49 is true, DCSC is loaded with the output of 4-BIT adder J-34 and some constant BITS. The resultant value is equal to (NDC - 49). Otherwise DCSC and NLO are indexed by one. Should DCSC reach - 0 the output multiplexer J-9 is set to output a constant binary 8 and the signal DCS (Data Channel Section) goes valid for the remainder of the frame.

When NLO reaches -0 a D-flop in package J-49 is set. The output of this D-flop is called ACT 1. ACT 1 in turn reenables ACT so that NLO will be indexed by the next CAS pulse. This will remove the carry out condition thatinitially set ACT 1. ACT 1 is itself gated by CAS and the result is used to enable LO to index from L to L + 1 during the CAS pulse following the initial carry out of NLO. The delay insures that LO is indexed after the NL'th active channel has been sampled. Note again that LO cannot exceed 8. During this time the output of LO is selected for output by multiplexer J-9. LO is outputed as the value of L until the DCSC counter reaches -0 as described previously.

The number of data channels selected by the thumbwheel switch one frame previously is output on the lines NDC1, NDC2, and NDC4 during the output frame.

In conclusion it has been shown how the Compute L transmit card takes in values representing the number of active voice channels and the number of data channels in a given frame. During the following frame the results of the computation of the previous frame are formatted for output to the multiplexer and to the predictive encoder.

4.13 Receiver Compute L Board (Board T)

The purpose of the receiver Compute L board is to compute the number of bits to be allocated to each active channel. The card also formats the results of the computation for output to the Bit Stream Decoder and to the Quantizer Decoder.

Computation Section

It is the job of the computation section to determine the number of bits that will be allocated to each active voice channel. Appendix D lists the number of bits assigned to the active voice channels as a function of the number of active voice channels and the number of data channels.

Notice that if the number of data and voice channels combined exceeds 22, a certain number of the voice channels receive from 3 to 7 bits per channel and the remaining voice channels receive 1 plus this number of bits (although sometimes all of the active channels receive the same number of bits). This scheme insures that all of the available bits in a frame are used.

The following definitions and equations describe the algorithm used to compute the number of bits assigned to each channel.

Input and Output Variables

NACH = Number of Active voice CHannels for next frame to be processed by the bit stream decoder (obtained from channel assignment.

NDC = Number of Data Channels (obtained from thumbwheel switch on Compute L board).

L = Number of bits allocated to first NL (see below) active voice channels.

NL = Number of active voice channels receiving L bits per channel (remaining (NACH - NL) channels receive (L + 1) bits per channel).

Internal Variables

BITSA = 180 - (NDC * 8) --- total number of BITS vailable to the active voice channels.

FILL = Number of unused bits in the frame.

ACC = Active Channel Counter

BITC = BITs available Counter

FILLI = Internal FILL latch

LI = Value of L Internal

Equations used by Compute L

L = MIN (8, BITSA/NACH)

NL = Remainder of BITSA/NACH Division (or equal to NACH if remainder is zero).

FILL = MAX (0, BITSA -(NACH *8))

Fig. 4.53 is a block diagram which illustrates the structure of the Compute L card. During D9, BITC (counters T-39 and T-46) are loaded with -BITSA (1's complement). This value is obtained from the thumb-wheel switch and the 4 bit adder T-30. Also ACC (Counters T-3 and T-11) is loaded with -NACH (1's complement) and LI (counter T-35 and T-11) is set to zero (exception: LI is set to 8 if NACH = zero).

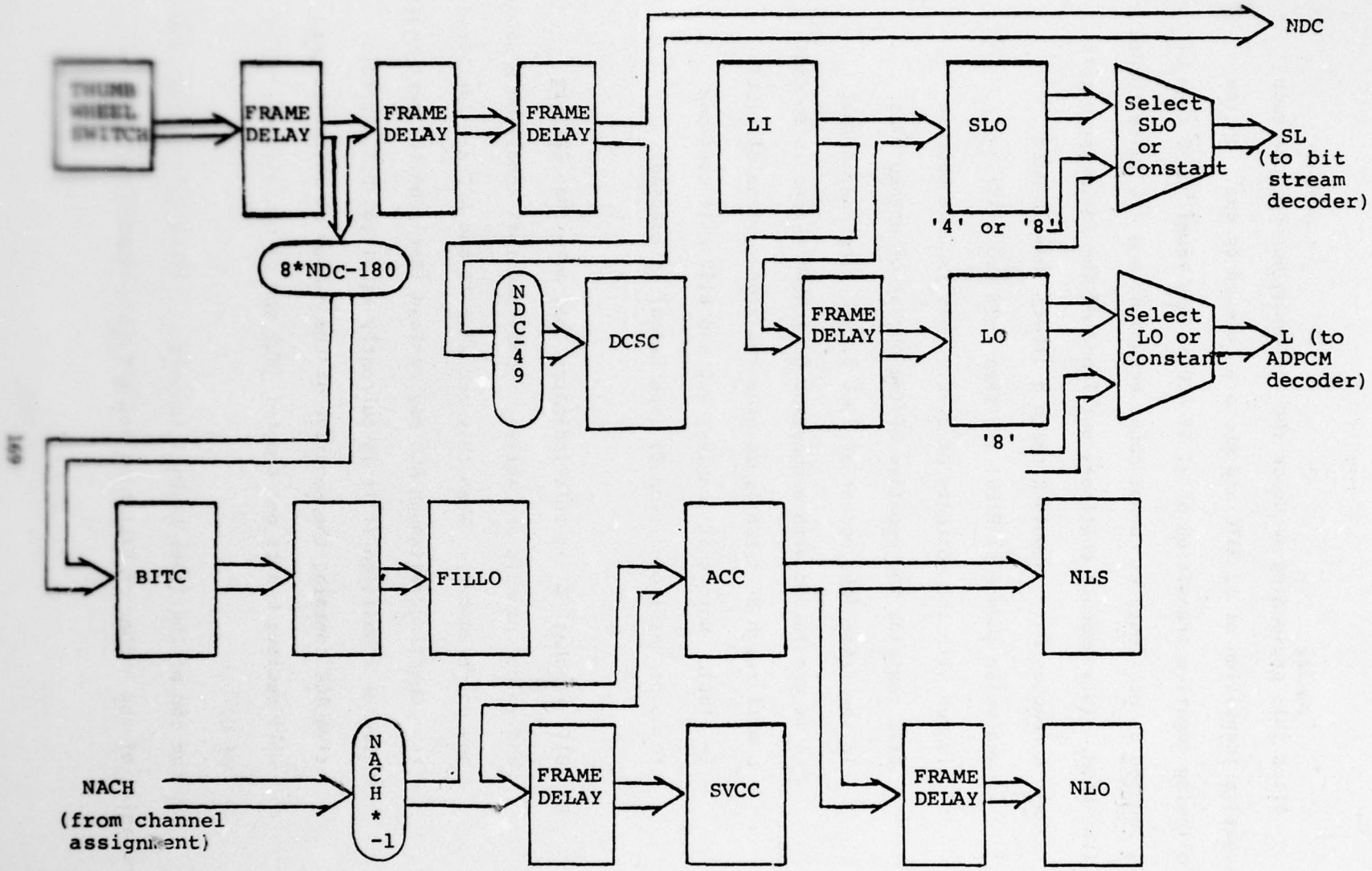


Figure 4.53 Major data paths for receive compute L board

After this preloading sequence the following occurs. On each negative transition of $\phi 2$, BITC and ACC are indexed by one. On the following negative transition of $\phi 1$ if ACC is not equal to -0 then LI is indexed by one (note LI cannot count beyond 8) and ACC is re-loaded with -NACH. This sequence continues until one of the following occurs:

A) L reaches 8 --- should L reach 8 (fewer than 22 data and voice channels) FILLI (latches T-37 and T-45) are loaded with the contents of BITC. This is done because BITC contains the negative of the number of unused bits in the frame (the number of FILL bits). The reason that LI is pre-loaded with a constant 8 if NACH = zero is so that L will reach 8 instantly to give the proper value of FILLI (= -180). Note that counters ACC and BITC will continue to count until condition B) (see below) is reached.

B) BITC reaches -0 --- this indicates that more than 22 data and voice channels are active (provided that condition A above has not been met). When this occurs BITC and ACC are inhibited from counting (although ACC may re-load itself on the next negative $\phi 1$ transition if it is currently equal to -0). At this time ACC contains the negative of the number of channels that will recieve L bits per channel (NL) and LI contains the value of L.

By the end of the frame LI will contain L. FILLI will contain the negative of the number of FILL bits and ACC will contain -NL.

Output Section

During D9 the variables FILLI, LI, and NL from the last frames computation are contained in the computational section. During the negative transition of D9 anded with $\phi 2$ these values are transferred to the output registers. The following paragraphs describe how these values are passed on to the bit stream decoder.

Bit Stream Decoder

The Bit Stream Decoder requires the 4 bit binary value of L (called SL) for the channel it is deserializing and a signal called -FILLN which is low to indicate that the channel the Bit Stream Decoder is deserializing is a non-active channel (FILL). The bit stream decoder supplies a narrow active high clock pulse called RCLK to indicate that it has sampled SL and -FILLN. This pulse is extended out by the D-flops in package T-1 to be one slow clock period long. This clock is used to update the output control counters for SL and -FILLN. This extended pulse is called RCLKB.

During D9 counter FILLO (FILL Output counter -- packages T-36 and T-44) is loaded with - (number of FILL bits/4). This value is present in FILLI at this time. Also NLS (Number at L serial --- counters T-20 and T-28) is loaded with -NL (present in ACC at this time). The value of L is also clocked into SLO (Counter T-34) at this time (obtained from LI).

During the trailing edge of D9 6-bit latch T-26 is clocked. This effectively delays -NACH by one frame. This delayed -NACH is loaded into SVCC (Serial Voice Channel Counter -- packages T-19 and T-27) on the next negative transition of (D9 & $\phi 1$).

On the trailing edge of D9 D-flop T-51 is clocked. Its' data input is tied to the most significant bit of SLO. The Q-not output of this flop provides the -FILLN signal which indicates that there are some FILL bits in the frame. When FILLN is set, the following occurs. During the negative transition of ϕ2 anded with RCLKB counter FILLO is indexed. If the result is less than -0 nothing occurs. When FILLO reaches -0 the carry out of this counter resets FILLN. Note that the FILLN signal causes the 2 to 1 multiplexer in package T-33 to select a constant binary 4 to output to the Bit Stream Decoder. Since FILLO contains the value -(FILL/4) this results in the bit stream decoder ignoring the first (FILL)bits in the frame.

After FILLN goes false (or at the start of the frame if FILLN was false initially) the multiplexer is addressed to look at the output of SLO. SLO initially contains the value of L (see above). After the negative transition of RCLKB anded with ¢2 anded with -FILLN the NLS and the SVCC counters are indexed by 1. If the result in the NLS counter is less than -0 nothing occurs. Should NLS reach -0 the SLO counter is indexed by 1 (it cannot exceed 8 however) and this becomes the new value of SL (equal to L + 1).

The multiplexer continues to output the value in the SLO counter until SVCC reaches -0 (this will always occur in a frame since there are always 49 RCLK pulses in a frame). Remember that SVCC initially contained the value -NACH for this output frame. Therefore when SVCC reaches -0 this signifies that all the voice channels have been processed and the remaining channels are data channels. The carry out signal of SVCC forces the multiplexer to select a constant binary 8. This is

done to insure that all data channels are allocated 8 bits. The constant 8 is output until the start of the next frame when the above process repeats itself.

Quantizer Decoder

During the frame following the outputting of SL to the Bit Stream Decoder the same L (same means that the same number of active voice channels recieve L bits) is output to the Quantizer Decoder. In order to do this however the results of the initial computation must be stored for an entire frame and then transfered to the output stage. This delay for L is accomplished by counter T-43 (actually used just as a gated edge triggered latch) and delay for NL is handled by counters T-4 and T-12 (this is currently unnecessary but is needed if the NDC information is transmitted in the serial bit stream at some time in the future).

The output stages for this section are loaded by the negative transition of (D9 and ¢2). The counters loaded by this condition are LO (L Output -- counter T-42) and NLO (Number at L Output -- counters T-5 and T-13). D9 also sets a D-flop in package T-53 which allows the loading of DCSC (Data Channel Start Counter -- counters T-21 and T-29).

The basic clock signal for this section is generated by the D-flops in package T-55 and is called CAS (Channel Acquisition Strobe). There are 48 CAS pulses in a frame and they are 4 slow clock pulses apart. When CAS is valid, a signal called -RACT (Receive Active Channel) is supplied to the card by channel assignment. -RACT is active low to indicate that the channel slot being "looked at" is to be occupied by an active voice channel.

During CAS if -RACT is high nothing occurs. If however - RACT is LOW the following occurs. If D-flop T-53 is true, DCSC is loaded with the output of 4-bit adder T-23 and some constant bits. The resultant value is equal to (NDC - 49). Otherwise DCSC and NLO are indexed by one. Should DCSC reach -0 the output multiplexor T-41 is set to output a constant binary 8 and the signal DCS (Data Channel Section) goes valid for the remainder of the frame.

When NLO reaches -0 a D-flop in package T-53 is set. The output of this D-flop is called ACT 1. ACT 1 in turn re-enables ACT so that NLO will be indexed by the next CAS pulse. In this way, the carry out condition that initially set ACT 1 will be removed. ACT 1 is itself gated by CAS and the result is used to enable LO to index from L to L + 1 during the CAS pulse following the initial carry out of NLO. This one CAS period delay of NLO's carry out is necessary because the first CAS in a frame is used solely to load DCSC. The delay insures that LO is indexed after the NL'th active channel has been sampled. Note again that LO cannot exceed 8. During this time the output of LO is selected for output by multiplexer T-41. LO is output as the value of L until the DCSC counter reaches -0 as described previously.

The number of data channels selected by the thumbwheel switch two frames previously is output on the lines NDC1, NDC2, and NDC4 during the output frame.

In conclusion it has been shown how the Compute L card takes in values representing the number of active voice channels and the number of data channels in a given frame. During the following frame the results of the previous frame's computation are output to the Bit Stream Decoder. During the frame following this (2nd frame after NACH and NDC were used for computation) the same results of the initial computation are output to the Quantizer Decoder.

APPENDIX A - Range distribution

Range Number (Decimal)	Range Size (Decimal)	Upper Range Limit (Octal)	Range Reciprocal (Octal)
0	144	40110	1.62000
1	152	40114	1.54000
2	164	40122	1.44000
3	184	40134	1.31000
4	200	40144	1.22000
5	216	40154	1.14000
6	232	40164	1.06400
7	256	40200	1.00000
8	288	40220	0.71000
9	304	40230	0.66000
10	328	40244	0.62000
11	368	40270	0.54400
12	400	40310	0.51000
13	432	40330	0.46000
14	464	40350	0.43200
15	512	40400	0.40000
16	576	40440	0.34400
17	608	40460	0.33000
18	656	40510	0.31000
19	736	40560	0.26200
20	800	40620	0.24400
21	864	40660	0.23000
22	928	40720	0.21500
23	1024	41000	0.20000
24	1152	41100	0.16200
25	1216	41140	0.15400
26	1312	41220	0.14400
27	1472	41340	0.13100
28	1600	41440	0.12200
29	1728	41540	0.11400
30	1856	41640	0.10640
31	2048	42000	0.10000
32	2304	42200	0.07100
33	2432	42300	0.06600
34	2624	42440	0.06200
35	2933	42700	0.05440
36	3200	43100	0.05100
37	3456	43300	0.04600
38	3712	43500	0.04320
39	4096	44000	0.04000
40	4608	44400	0.03440
41	4864	44600	0.03300
42	5248	45100	0.03100
43	5888	45600	0.02620
44	6400	46200	0.02440
45	6912	46600	0.02300
46	7424	47200	0.02150
47	8192	50000	0.02000
48	9216	51000	0.01620
49	9728	51400	0.01540
50	10496	52200	0.01440
51	11776	53400	0.01310
52	12800	54400	0.01220
53	13824	55400	0.01140
54	14848	56400	0.01064
55	16384	60000	0.01000
56	18432	62000	0.00710
57	19456	66000	0.00660
58	20992	64400	0.00620
59	23552	67000	0.00544
60	25600	71000	0.00510
61	27648	73000	0.00460
62	29696	75000	0.00432

APPENDIX B - The adaptation strategy

Output Word	Range Number Adaptation
0 - 6	+14
7 - 13	+13
14 - 19	+12
20 - 25	+11
26 - 31	+10
32 - 36	+ 9
37 - 41	+ 8
42 - 45	+ 7
46 - 49	+ 6
50 - 52	+ 5
53 - 55	+ 4
56 - 58	+ 3
59 - 61	+ 2
62 - 65	+ 1
66 - 190	- 1
191 - 194	+ 1
195 - 197	+ 2
198 - 200	+ 3
201 - 203	+ 4
204 - 206	+ 5
207 - 210	+ 6
211 - 214	+ 7
215 - 219	+ 8
220 - 224	+ 9
225 - 230	+10
231 - 236	+11
237 - 242	+12
243 - 249	+13
250 - 255	+14

APPENDIX C

μ255-to-linear Code Conversion Table for PROMS

The following tables were generated using an AGT 30 computer. The PROM d1 contains the 8 least significant bits of the two's complement linear output while PROM d3 contains the 6 most significant bits.

```
PROMD1     VER 1    REV      12 OCT 77

1
2       * TABLE FOR U255 TO LINEAR CONVERSION (OCTAL)
3        82S114  OCTD  OCTA
4       * THE TABLE CONTAINS THE OCTAL ADDRESS  ,INVERTED PCM,
5       * FOLLOWED BY TWOS COMPLEMENT DATA
6       * THEN UNINVERTED PCM AND YHAT+33 PER GREY BOOK P411
7       * FIRST IN OCTAL THEN IN DECIMAL
10
11      NEW TABLE
12      377:000       *000:00041          0            33
13      376:002       *001:00043          1            35
14      375:004       *002:00045          2            37
15      374:006       *003:00047          3            39
16      373:010       *004:00051          4            41
17      372:012       *005:00053          5            43
20      371:014       *006:00055          6            45
21      370:016       *007:00057          7            47
22      367:020       *010:00061          8            49
23      366:022       *011:00063          9            51
24      365:024       *012:00065         10            53
25      364:026       *013:00067         11            55
26      363:030       *014:00071         12            57
27      362:032       *015:00073         13            59
30      361:034       *016:00075         14            61
31      360:036       *017:00077         15            63
32      357:041       *020:00102         16            66
33      356:045       *021:00106         17            70
34      355:051       *022:00112         18            74
35      354:055       *023:00116         19            78
36      353:061       *024:00122         20            82
37      352:065       *025:00126         21            86
40      351:071       *026:00132         22            90
41      350:075       *027:00136         23            94
42      347:101       *030:00142         24            98
43      346:105       *031:00146         25           102
44      345:111       *032:00152         26           106
45      344:115       *033:00156         27           110
46      343:121       *034:00162         28           114
47      342:125       *035:00166         29           118
50      341:131       *036:00172         30           122
51      340:135       *037:00176         31           126
52      337:143       *040:00204         32           132
53      336:153       *041:00214         33           140
54      335:163       *042:00224         34           148
55      334:173       *043:00234         35           156
56      333:203       *044:00244         36           164
57      332:213       *045:00254         37           172
60      331:223       *046:00264         38           180
61      330:233       *047:00274         39           188
62      327:243       *050:00304         40           196
63      326:253       *051:00314         41           204
64      325:263       *052:00324         42           212
65      324:273       *053:00334         43           220
66      323:303       *054:00344         44           228
```

PROMD1 VER 1 REV 12 OCT 77

1	322:313	*055:00354	45	236
2	321:323	*056:00364	46	244
3	320:333	*057:00374	47	252
4	317:347	*060:00410	48	264
5	316:367	*061:00430	49	280
6	315:007	*062:00450	50	296
7	314:027	*063:00470	51	312
10	313:047	*064:00510	52	328
11	312:067	*065:00530	53	344
12	311:107	*066:00550	54	360
13	310:127	*067:00570	55	376
14	307:147	*070:00610	56	392
15	306:167	*071:00630	57	408
16	305:207	*072:00650	58	424
17	304:227	*073:00670	59	440
20	303:247	*074:00710	60	456
21	302:267	*075:00730	61	472
22	301:307	*076:00750	62	488
23	300:327	*077:00770	63	504
24	277:357	*100:01020	64	528
25	276:017	*101:01060	65	560
26	275:057	*102:01120	66	592
27	274:117	*103:01160	67	624
30	273:157	*104:01220	68	656
31	272:217	*105:01260	69	688
32	271:257	*106:01320	70	720
33	270:317	*107:01360	71	752
34	267:357	*110:01420	72	784
35	266:017	*111:01460	73	816
36	265:057	*112:01520	74	848
37	264:117	*113:01560	75	880
40	263:157	*114:01620	76	912
41	262:217	*115:01660	77	944
42	261:257	*116:01720	78	976
43	260:317	*117:01760	79	1008
44	257:377	*120:02040	80	1056
45	256:077	*121:02140	81	1120
46	255:177	*122:02240	82	1184
47	254:277	*123:02340	83	1248
50	253:377	*124:02440	84	1312
51	252:077	*125:02540	85	1376
52	251:177	*126:02640	86	1440
53	250:277	*127:02740	87	1504
54	247:377	*130:03040	88	1568
55	246:077	*131:03140	89	1632
56	245:177	*132:03240	90	1696
57	244:277	*133:03340	91	1760
60	243:377	*134:03440	92	1824
61	242:077	*135:03540	93	1888
62	241:177	*136:03640	94	1952
63	240:277	*137:03740	95	2016
64	237:037	*140:04100	96	2112
65	236:237	*141:04300	97	2240
66	235:037	*142:04500	98	2368

PROMD1 VER 1 REV 12 OCT 77

1	234:237	*143:04700	99	2496
2	233:037	*144:05100	100	2624
3	232:237	*145:05300	101	2752
4	231:037	*146:05500	102	2880
5	230:237	*147:05700	103	3008
6	227:037	*150:06100	104	3136
7	226:237	*151:06300	105	3264
10	225:037	*152:06500	106	3392
11	224:237	*153:06700	107	3520
12	223:037	*154:07100	108	3648
13	222:237	*155:07300	109	3776
14	221:037	*156:07500	110	3904
15	220:237	*157:07700	111	4032
16	217:137	*160:10200	112	4224
17	216:137	*161:10600	113	4480
20	215:137	*162:11200	114	4736
21	214:137	*163:11600	115	4992
22	213:137	*164:12200	116	5248
23	212:137	*165:12600	117	5504
24	211:137	*166:13200	118	5760
25	210:137	*167:13600	119	6016
26	207:137	*170:14200	120	6272
27	206:137	*171:14600	121	6528
30	205:137	*172:15200	122	6784
31	204:137	*173:15600	123	7040
32	203:137	*174:16200	124	7296
33	202:137	*175:16600	125	7552
34	201:137	*176:17200	126	7808
35	200:137	*177:17600	127	8064
36	177:377	*200:00041	128	33
37	176:376	*201:00043	129	35
40	175:374	*202:00045	130	37
41	174:372	*203:00047	131	39
42	173:370	*204:00051	132	41
43	172:366	*205:00053	133	43
44	171:364	*206:00055	134	45
45	170:362	*207:00057	135	47
46	167:360	*210:00061	136	49
47	166:356	*211:00063	137	51
50	165:354	*212:00065	138	53
51	164:352	*213:00067	139	55
52	163:350	*214:00071	140	57
53	162:346	*215:00073	141	59
54	161:344	*216:00075	142	61
55	160:342	*217:00077	143	63
56	157:337	*220:00102	144	66
57	156:333	*221:00106	145	70
60	155:327	*222:00112	146	74
61	154:323	*223:00116	147	78
62	153:317	*224:00122	148	82
63	152:313	*225:00126	149	86
64	151:307	*226:00132	150	90
65	150:303	*227:00136	151	94
66	147:277	*230:00142	152	98

PROMD1 VER 1 REV 12 OCT 77

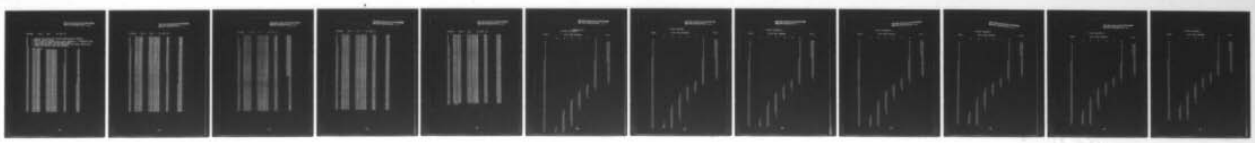
1	146:273	*231:00146	153	102
2	145:267	*232:00152	154	106
3	144:263	*233:00156	155	110
4	143:257	*234:00162	156	114
5	142:253	*235:00166	157	118
6	141:247	*236:00172	158	122
7	140:243	*237:00176	159	126
10	137:235	*240:00204	160	132
11	136:225	*241:00214	161	140
12	135:215	*242:00224	162	148
13	134:205	*243:00234	163	156
14	133:175	*244:00244	164	164
15	132:165	*245:00254	165	172
16	131:155	*246:00264	166	180
17	130:145	*247:00274	167	188
20	127:135	*250:00304	168	196
21	126:125	*251:00314	169	204
22	125:115	*252:00324	170	212
23	124:105	*253:00334	171	220
24	123:075	*254:00344	172	228
25	122:065	*255:00354	173	236
26	121:055	*256:00364	174	244
27	120:045	*257:00374	175	252
30	117:031	*260:00410	176	264
31	116:011	*261:00430	177	280
32	115:371	*262:00450	178	296
33	114:351	*263:00470	179	312
34	113:331	*264:00510	180	328
35	112:311	*265:00530	181	344
36	111:271	*266:00550	182	360
37	110:251	*267:00570	183	376
40	107:231	*270:00610	184	392
41	106:211	*271:00630	185	408
42	105:171	*272:00650	186	424
43	104:151	*273:00670	187	440
44	103:131	*274:00710	188	456
45	102:111	*275:00730	189	472
46	101:071	*276:00750	190	488
47	100:051	*277:00770	191	504
50	077:021	*300:01020	192	528
51	076:361	*301:01060	193	560
52	075:321	*302:01120	194	592
53	074:261	*303:01160	195	624
54	073:221	*304:01220	196	656
55	072:161	*305:01260	197	688
56	071:121	*306:01320	198	720
57	070:061	*307:01360	199	752
60	067:021	*310:01420	200	784
61	066:361	*311:01460	201	816
62	065:321	*312:01520	202	848
63	064:261	*313:01560	203	880
64	063:221	*314:01620	204	912
65	062:161	*315:01660	205	944
66	061:121	*316:01720	206	976

```
PROMD1      VER 1     REV       12 OCT 77

1       060:061       *317:01760       207          1008
2       057:001       *320:02040       208          1056
3       056:301       *321:02140       209          1120
4       055:201       *322:02240       210          1184
5       054:101       *323:02340       211          1248
6       053:001       *324:02440       212          1312
7       052:301       *325:02540       213          1376
10      051:201       *326:02640       214          1440
11      050:101       *327:02740       215          1504
12      047:001       *330:03040       216          1568
13      046:301       *331:03140       217          1632
14      045:201       *332:03240       218          1696
15      044:101       *333:03340       219          1760
16      043:001       *334:03440       220          1824
17      042:301       *335:03540       221          1888
20      041:201       *336:03640       222          1952
21      040:101       *337:03740       223          2016
22      037:341       *340:04100       224          2112
23      036:141       *341:04300       225          2240
24      035:341       *342:04500       226          2368
25      034:141       *343:04700       227          2496
26      033:341       *344:05100       228          2624
27      032:141       *345:05300       229          2752
30      031:341       *346:05500       230          2880
31      030:141       *347:05700       231          3008
32      027:341       *350:06100       232          3136
33      026:141       *351:06300       233          3264
34      025:341       *352:06500       234          3392
35      024:141       *353:06700       235          3520
36      023:341       *354:07100       236          3648
37      022:141       *355:07300       237          3776
40      021:341       *356:07500       238          3904
41      020:141       *357:07700       239          4032
42      017:241       *360:10200       240          4224
43      016:241       *361:10600       241          4480
44      015:241       *362:11200       242          4736
45      014:241       *363:11600       243          4992
46      013:241       *364:12200       244          5248
47      012:241       *365:12600       245          5504
50      011:241       *366:13200       246          5760
51      010:241       *367:13600       247          6016
52      007:241       *370:14200       248          6272
53      006:241       *371:14600       249          6528
54      005:241       *372:15200       250          6784
55      004:241       *373:15600       251          7040
56      003:241       *374:16200       252          7296
57      002:241       *375:16600       253          7552
60      001:241       *376:17200       254          7808
61      000:241       *377:17600       255          8064
62        END
```

AD-A056 986 GEORGIA INST OF TECH ATLANTA F/G 17/2
DIGROUP DATA REDUCTION SYSTEM.(U)
JUN 78 J B O'NEAL
UNCLASSIFIED RADC-TR-78-139 F30602-75-C-0118 NL

3 OF 3
AD
A056986



END
DATE
FILMED
9-78
DDC

```
PROMD3     VER 1    REV      12 OCT 77

1
2       * TABLE FOR U255 TO LINEAR CONVERSION (OCTAL)
3         82S114   OCTD   OCTA
4       * THE TABLE CONTAINS THE OCTAL ADDRESS  ,INVERTED PCM,
5       * FOLLOWED BY TWOS COMPLEMENT DATA
6       * THEN UNINVERTED PCM AND YHAT+33 PER GREY BOOK P411
7       * FIRST IN OCTAL THEN IN DECIMAL
10
11      NEW TABLE
12      377:000         *000:00041          0            33
13      376:000         *001:00043          1            35
14      375:000         *002:00045          2            37
15      374:000         *003:00047          3            39
16      373:000         *004:00051          4            41
17      372:000         *005:00053          5            43
20      371:000         *006:00055          6            45
21      370:000         *007:00057          7            47
22      367:000         *010:00061          8            49
23      366:000         *011:00063          9            51
24      365:000         *012:00065         10            53
25      364:000         *013:00067         11            55
26      363:000         *014:00071         12            57
27      362:000         *015:00073         13            59
30      361:000         *016:00075         14            61
31      360:000         *017:00077         15            63
32      357:000         *020:00102         16            66
33      356:000         *021:00106         17            70
34      355:000         *022:00112         18            74
35      354:000         *023:00116         19            78
36      353:000         *024:00122         20            82
37      352:000         *025:00126         21            86
40      351:000         *026:00132         22            90
41      350:000         *027:00136         23            94
42      347:000         *030:00142         24            98
43      346:000         *031:00146         25           102
44      345:000         *032:00152         26           106
45      344:000         *033:00156         27           110
46      343:000         *034:00162         28           114
47      342:000         *035:00166         29           118
50      341:000         *036:00172         30           122
51      340:000         *037:00176         31           126
52      337:000         *040:00204         32           132
53      336:000         *041:00214         33           140
54      335:000         *042:00224         34           148
55      334:000         *043:00234         35           156
56      333:000         *044:00244         36           164
57      332:000         *045:00254         37           172
60      331:000         *046:00264         38           180
61      330:000         *047:00274         39           188
62      327:000         *050:00304         40           196
63      326:000         *051:00314         41           204
64      325:000         *052:00324         42           212
65      324:000         *053:00334         43           220
66      323:000         *054:00344         44           228
```

PROMD3 VER 1 REV 12 OCT 77

1	322:000	*055:00354	45	236
2	321:000	*056:00364	46	244
3	320:000	*057:00374	47	252
4	317:000	*060:00410	48	264
5	316:000	*061:00430	49	280
6	315:001	*062:00450	50	296
7	314:001	*063:00470	51	312
10	313:001	*064:00510	52	328
11	312:001	*065:00530	53	344
12	311:001	*066:00550	54	360
13	310:001	*067:00570	55	376
14	307:001	*070:00610	56	392
15	306:001	*071:00630	57	408
16	305:001	*072:00650	58	424
17	304:001	*073:00670	59	440
20	303:001	*074:00710	60	456
21	302:001	*075:00730	61	472
22	301:001	*076:00750	62	488
23	300:001	*077:00770	63	504
24	277:001	*100:01020	64	528
25	276:002	*101:01060	65	560
26	275:002	*102:01120	66	592
27	274:002	*103:01160	67	624
30	273:002	*104:01220	68	656
31	272:002	*105:01260	69	688
32	271:002	*106:01320	70	720
33	270:002	*107:01360	71	752
34	267:002	*110:01420	72	784
35	266:003	*111:01460	73	816
36	265:003	*112:01520	74	848
37	264:003	*113:01560	75	880
40	263:003	*114:01620	76	912
41	262:003	*115:01660	77	944
42	261:003	*116:01720	78	976
43	260:003	*117:01760	79	1008
44	257:003	*120:02040	80	1056
45	256:004	*121:02140	81	1120
46	255:004	*122:02240	82	1184
47	254:004	*123:02340	83	1248
50	253:004	*124:02440	84	1312
51	252:005	*125:02540	85	1376
52	251:005	*126:02640	86	1440
53	250:005	*127:02740	87	1504
54	247:005	*130:03040	88	1568
55	246:006	*131:03140	89	1632
56	245:006	*132:03240	90	1696
57	244:006	*133:03340	91	1760
60	243:006	*134:03440	92	1824
61	242:007	*135:03540	93	1888
62	241:007	*136:03640	94	1952
63	240:007	*137:03740	95	2016
64	237:010	*140:04100	96	2112
65	236:010	*141:04300	97	2240
66	235:011	*142:04500	98	2368

PROMD3 VER 1 REV 12 OCT 77

1	234:011	*143:04700	99	2496
2	233:012	*144:05100	100	2624
3	232:012	*145:05300	101	2752
4	231:013	*146:05500	102	2880
5	230:013	*147:05700	103	3008
6	227:014	*150:06100	104	3136
7	226:014	*151:06300	105	3264
10	225:015	*152:06500	106	3392
11	224:015	*153:06700	107	3520
12	223:016	*154:07100	108	3648
13	222:016	*155:07300	109	3776
14	221:017	*156:07500	110	3904
15	220:017	*157:07700	111	4032
16	217:020	*160:10200	112	4224
17	216:021	*161:10600	113	4480
20	215:022	*162:11200	114	4736
21	214:023	*163:11600	115	4992
22	213:024	*164:12200	116	5248
23	212:025	*165:12600	117	5504
24	211:026	*166:13200	118	5760
25	210:027	*167:13600	119	6016
26	207:030	*170:14200	120	6272
27	206:031	*171:14600	121	6528
30	205:032	*172:15200	122	6784
31	204:033	*173:15600	123	7040
32	203:034	*174:16200	124	7296
33	202:035	*175:16600	125	7552
34	201:036	*176:17200	126	7808
35	200:037	*177:17600	127	8064
36	177:077	*200:00041	128	33
37	176:077	*201:00043	129	35
40	175:077	*202:00045	130	37
41	174:077	*203:00047	131	39
42	173:077	*204:00051	132	41
43	172:077	*205:00053	133	43
44	171:077	*206:00055	134	45
45	170:077	*207:00057	135	47
46	167:077	*210:00061	136	49
47	166:077	*211:00063	137	51
50	165:077	*212:00065	138	53
51	164:077	*213:00067	139	55
52	163:077	*214:00071	140	57
53	162:077	*215:00073	141	59
54	161:077	*216:00075	142	61
55	160:077	*217:00077	143	63
56	157:077	*220:00102	144	66
57	156:077	*221:00106	145	70
60	155:077	*222:00112	146	74
61	154:077	*223:00116	147	78
62	153:077	*224:00122	148	82
63	152:077	*225:00126	149	86
64	151:077	*226:00132	150	90
65	150:077	*227:00136	151	94
66	147:077	*230:00142	152	98

PROMD3 VER 1 REV 12 OCT 77

1	146:077	*231:00146	153	102
2	145:077	*232:00152	154	106
3	144:077	*233:00156	155	110
4	143:077	*234:00162	156	114
5	142:077	*235:00166	157	118
6	141:077	*236:00172	158	122
7	140:077	*237:00176	159	126
10	137:077	*240:00204	160	132
11	136:077	*241:00214	161	140
12	135:077	*242:00224	162	148
13	134:077	*243:00234	163	156
14	133:077	*244:00244	164	164
15	132:077	*245:00254	165	172
16	131:077	*246:00264	166	180
17	130:077	*247:00274	167	188
20	127:077	*250:00304	168	196
21	126:077	*251:00314	169	204
22	125:077	*252:00324	170	212
23	124:077	*253:00334	171	220
24	123:077	*254:00344	172	228
25	122:077	*255:00354	173	236
26	121:077	*256:00364	174	244
27	120:077	*257:00374	175	252
30	117:077	*260:00410	176	264
31	116:077	*261:00430	177	280
32	115:076	*262:00450	178	296
33	114:076	*263:00470	179	312
34	113:076	*264:00510	180	328
35	112:076	*265:00530	181	344
36	111:076	*266:00550	182	360
37	110:076	*267:00570	183	376
40	107:076	*270:00610	184	392
41	106:076	*271:00630	185	408
42	105:076	*272:00650	186	424
43	104:076	*273:00670	187	440
44	103:076	*274:00710	188	456
45	102:076	*275:00730	189	472
46	101:076	*276:00750	190	488
47	100:076	*277:00770	191	504
50	077:076	*300:01020	192	528
51	076:075	*301:01060	193	560
52	075:075	*302:01120	194	592
53	074:075	*303:01160	195	624
54	073:075	*304:01220	196	656
55	072:075	*305:01260	197	688
56	071:075	*306:01320	198	720
57	070:075	*307:01360	199	752
60	067:075	*310:01420	200	784
61	066:074	*311:01460	201	816
62	065:074	*312:01520	202	848
63	064:074	*313:01560	203	880
64	063:074	*314:01620	204	912
65	062:074	*315:01660	205	944
66	061:074	*316:01720	206	976

```
PROMD3     VER 1    REV      12 OCT 77

1      060:074      *317:01760      207      1008
2      057:074      *320:02040      208      1056
3      056:073      *321:02140      209      1120
4      055:073      *322:02240      210      1184
5      054:073      *323:02340      211      1248
6      053:073      *324:02440      212      1312
7      052:072      *325:02540      213      1376
10     051:072      *326:02640      214      1440
11     050:072      *327:02740      215      1504
12     047:072      *330:03040      216      1568
13     046:071      *331:03140      217      1632
14     045:071      *332:03240      218      1696
15     044:071      *333:03340      219      1760
16     043:071      *334:03440      220      1824
17     042:070      *335:03540      221      1888
20     041:070      *336:03640      222      1952
21     040:070      *337:03740      223      2016
22     037:067      *340:04100      224      2112
23     036:067      *341:04300      225      2240
24     035:066      *342:04500      226      2368
25     034:066      *343:04700      227      2496
26     033:065      *344:05100      228      2624
27     032:065      *345:05300      229      2752
30     031:064      *346:05500      230      2880
31     030:064      *347:05700      231      3008
32     027:063      *350:06100      232      3136
33     026:063      *351:06300      233      3264
34     025:062      *352:06500      234      3392
35     024:062      *353:06700      235      3520
36     023:061      *354:07100      236      3648
37     022:061      *355:07300      237      3776
40     021:060      *356:07500      238      3904
41     020:060      *357:07700      239      4032
42     017:057      *360:10200      240      4224
43     016:056      *361:10600      241      4480
44     015:055      *362:11200      242      4736
45     014:054      *363:11600      243      4992
46     013:053      *364:12200      244      5248
47     012:052      *365:12600      245      5504
50     011:051      *366:13200      246      5760
51     010:050      *367:13600      247      6016
52     007:047      *370:14200      248      6272
53     006:046      *371:14600      249      6528
54     005:045      *372:15200      250      6784
55     004:044      *373:15600      251      7040
56     003:043      *374:16200      252      7296
57     002:042      *375:16600      253      7552
60     001:041      *376:17200      254      7808
61     000:040      *377:17600      255      8064
62       END
```

Appendix D

0 DATA CHANNELS

NACH	BITS PER CHANNEL						FILL
	3	4	5	6	7	8	
0						0	180
1						1	172
2						2	164
3						3	156
4						4	148
5						5	140
6						6	132
7						7	124
8						8	116
9						9	108
10						10	100
11						11	92
12						12	84
13						13	76
14						14	68
15						15	60
16						16	52
17						17	44
18						18	36
19						19	28
20						20	20
21						21	12
22						22	4
23					4	19	
24					12	12	
25					20	5	
26				2	24		
27				9	18		
28				16	12		
29				23	6		
30				30	0		
31			6	25			
32			12	20			
33			18	15			
34			24	10			
35			30	5			
36			36	0			
37		5	32				
38		10	28				
39		15	24				
40		20	20				
41		25	16				
42		30	12				
43		35	8				
44		40	4				
45		45	0				
46	4	42					
47	8	39		188			
48	12	36					

1 DATA CHANNEL

NACH	BITS PER CHANNEL						FILL
	3	4	5	6	7	8	
0						0	172
1						1	164
2						2	156
3						3	148
4						4	140
5						5	132
6						6	124
7						7	116
8						8	108
9						9	100
10						10	92
11						11	84
12						12	76
13						13	68
14						14	60
15						15	52
16						16	44
17						17	36
18						18	28
19						19	20
20						20	12
21						21	4
22					4	18	
23					12	11	
24					20	4	
25				3	22		
26				10	16		
27				17	10		
28				24	4		
29			2	27			
30			8	22			
31			14	17			
32			20	12			
33			26	7			
34			32	2			
35		3	32				
36		8	28				
37		13	24				
38		18	20				
39		23	16				
40		28	12				
41		33	8				
42		38	4				
43		43	0				
44	4	40					
45	8	37					
46	12	34					
47	16	31					

2 DATA CHANNELS

NACH	BITS PER CHANNEL						FILL
	3	4	5	6	7	8	
0						0	164
1						1	156
2						2	148
3						3	140
4						4	132
5						5	124
6						6	116
7						7	108
8						8	100
9						9	92
10						10	84
11						11	76
12						12	68
13						13	60
14						14	52
15						15	44
16						16	36
17						17	28
18						18	20
19						19	12
20						20	4
21					4	17	
22					12	10	
23					20	3	
24				4	20		
25				11	14		
26				18	8		
27				25	2		
28			4	24			
29			10	19			
30			16	14			
31			22	9			
32			28	4			
33		1	32				
34		6	28				
35		11	24				
36		16	20				
37		21	16				
38		26	12				
39		31	8				
40		36	4				
41		41	0				
42	4	38					
43	8	35					
44	12	32					
45	16	29					
46	20	26					

3 DATA CHANNELS

NACH	BITS PER CHANNEL						FILL
	3	4	5	6	7	8	
0						0	156
1						1	148
2						2	140
3						3	132
4						4	124
5						5	116
6						6	108
7						7	100
8						8	92
9						9	84
10						10	76
11						11	68
12						12	60
13						13	52
14						14	44
15						15	36
16						16	28
17						17	20
18						18	12
19						19	4
20					4	16	
21					12	9	
22					20	2	
23				5	18		
24				12	12		
25				19	6		
26				26	0		
27			6	21			
28			12	16			
29			18	11			
30			24	6			
31			30	1			
32		4	28				
33		9	24				
34		14	20				
35		19	16				
36		24	12				
37		29	8				
38		34	4				
39		39	0				
40	4	36					
41	8	33					
42	12	30					
43	16	27					
44	20	24					
45	24	21					

4 DATA CHANNELS

NACH	BITS PER CHANNEL						FILL
	3	4	5	6	7	8	
0						0	148
1						1	140
2						2	132
3						3	124
4						4	116
5						5	108
6						6	100
7						7	92
8						8	84
9						9	76
10						10	68
11						11	60
12						12	52
13						13	44
14						14	36
15						15	28
16						16	20
17						17	12
18						18	4
19					4	15	
20					12	8	
21					20	1	
22				6	16		
23				13	10		
24				20	4		
25			2	23			
26			8	18			
27			14	13			
28			20	8			
29			26	3			
30		2	28				
31		7	24				
32		12	20				
33		17	16				
34		22	12				
35		27	8				
36		32	4				
37		37	0				
38	4	34					
39	8	31					
40	12	28					
41	16	25					
42	20	22					
43	24	19					
44	28	16					

5 DATA CHANNELS

NACH	BITS PER CHANNEL .						FILL
	3	4	5	6	7	8	
0						0	140
1						1	132
2						2	124
3						3	116
4						4	108
5						5	100
6						6	92
7						7	84
8						8	76
9						9	68
10						10	60
11						11	52
12						12	44
13						13	36
14						14	28
15						15	20
16						16	12
17						17	4
18					4	14	
19					12	7	
20					20	0	
21				7	14		
22				14	8		
23				21	2		
24			4	20			
25			10	15			
26			16	10			
27			22	5			
28			28	0			
29		5	24				
30		10	20				
31		15	16				
32		20	12				
33		25	8				
34		30	4				
35		35	0				
36	4	32					
37	8	29					
38	12	26					
39	16	23					
40	20	20					
41	24	17					
42	28	14					
43	32	11					

6 DATA CHANNELS

NACH	BITS PER CHANNEL						FILL
	3	4	5	6	7	8	
0						0	132
1						1	124
2						2	116
3						3	108
4						4	100
5						5	92
6						6	84
7						7	76
8						8	68
9						9	60
10						10	52
11						11	44
12						12	36
13						13	28
14						14	20
15						15	12
16						16	4
17					4	13	
18					12	6	
19				1	18		
20				8	12		
21				15	6		
22				22	0		
23			6	17			
24			12	12			
25			18	7			
26			24	2			
27		3	24				
28		8	20				
29		13	16				
30		18	12				
31		23	8				
32		28	4				
33		33	0				
34	4	30					
35	8	27					
36	12	24					
37	16	21					
38	20	18					
39	24	15					
40	28	12					
41	32	9					
42	36	6					